SIMULATION STUDY ON NEW SAMPLING METHOD WHEN DATA IS LIMITED

A SIMULATION STUDY TO EXAMINE A NEW METHOD OF SAMPLING WHEN INFORMATION ON THE TARGET POPULATION IS VERY LIMITED

By

ZAHRA AHSAN, B. MATH.

A Thesis

Submitted to the School of Graduate Studies

in Partial Fulfilment of the Requirements

for the Degree

Master of Science

McMaster University

MASTER OF SCIENCE (2011) McMaster University
(Statistics) Hamilton, Ontario

TITLE: A Simulation Study to Examine a New Method of Sampling When Information on the Target Population Is Very Limited

AUTHOR: Zahra Ahsan, B.Math. (University of Waterloo)

SUPERVISOR: Professor H. Shannon

NUMBER OF PAGES: x, 118

ABSTRACT

A new sampling method was developed for use in areas where standard methods for sampling from a population may be difficult or impossible to use. The new method uses Global Positioning System (GPS) points and satellite photos to identify datapoints (locations) within selected towns. A circle (of a specified radius) is drawn around each datapoint, which may include several buildings or sometimes none. A single household is sampled from each circle and then one adult (aged 18 years or older) is interviewed from the household, based on who had the most recent birthday. Two issues arise with this new sampling method: first, the probability of sampling any household may vary, depending on the datapoint its chosen from; and second, circles surrounding the chosen datapoints may overlap. The thesis used simulations to see whether the first issue affected the point estimates of two population parameters. The second issue was too complex to investigate so it was ignored when sampling households. Simulations were run to test the sampling method on two different hypothetical towns, one with denser population than the other. Results from the simulations showed that estimates did not always match what was expected, but the observed differences were not substantial. It was presumed that the reason for differences was due to the issue of multiple probabilities for selection as well as overlapping circles (which had been ignored during sampling), since both issues were inherent in this sampling method. Although some differences were observed from true values, they appeared not to be very different from the population values, so I conclude that this sampling method is a useful one. Future work may look more closely at understanding the issue of overlapping circles and how it affects the point estimates.

ACKNOWLEDGEMENTS

It is with great pleasure that I acknowledge the support of my thesis supervisor, Professor Harry Shannon, who guided me throughout these last few months as I worked on the thesis. I wish to thank him for all his help, time and assistance during the course of my thesis term. In addition, I would also like to thank my wonderful husband for all his support while I worked on the thesis, right after we got married. I could not have done this without him. Finally, a huge thank you to my parents for their blessings, prayers and continued support in every aspect of my academic life.

TABLE OF CONTENTS

LIST OF FIGURES & TABLES

1. INTRODUCTION

A. Background

Much of statistics entails inferences from samples, so it is crucial to understand the methods used in sampling. Sometimes limited information is available on the population from which a sample is drawn, making it hard to use typical random, stratified or cluster sampling methods in such areas. These methods require details of the target population and it may not be practical in terms of cost, time, labor and access to certain areas, especially with stratified sampling, where the population has to be divided into appropriate strata. It requires information on each member of the population, which may be unavailable, difficult or very expensive to get, even though this method gives the most precision with regards to estimation. An alternative method is cluster sampling, which is cheaper than stratified sampling and increases sampling efficiency, but at the expense of less precision.

In areas where limited information about the population is present, such as war zones and refugee camps, conventional sampling methods are difficult to apply. This creates a need for a sampling method that does not require details of target population (as needed with the mentioned methods), and provides suitable estimates of the population parameters without being too costly. Such a method exists, which uses Global Positioning System (GPS) points in order to randomly select sampling units. A circle is drawn around the GPS point on a satellite photograph, and one of the buildings in the circle is randomly chosen to be in the sample.

My thesis investigates this new method of sampling, which allows one to estimate the probability of selection of a particular sampling unit/event (e.g. probability of selecting a resident in a remote village) with little a priori knowledge of the population. Additionally, the methodology allows the estimation of parameters of interest and standard errors.

This new sampling method does not lead to a fixed probability of sampling any given unit. Accordingly, the implications of this on the estimates of the parameters of interest and standard error are discussed in this thesis, via the use of simulations. I will be working with a simpler version of the original sampling method used in a survey in South Lebanon; the simplifications made to the method will be identified below.

The objective of my thesis was to explore the impact of the GPS and circle method of sampling on the point estimates and standard errors of population parameters.

B. Actual Sampling Method

The field work in South Lebanon was organized by a Wayne State University professor and interviewers from a local survey organization were used to survey individuals. One purpose of the study was to identify the proportion of South Lebanon's population who suffered some violation resulting from and following the fighting with Israel in 2006. In principle, the sampling method could have been used in any other place.

In this method, households were identified using Global Positioning System (GPS) points and satellite photos. When collecting the sample, 53 towns[1] in total were sampled. These included the 3 largest towns in South Lebanon in addition to 50 other towns that were selected using Probability Proportional to Estimated Size (PPES). Using satellite maps of the selected towns, GPS coordinates were randomly chosen to identify locations within those towns. A circle was then drawn around each GPS location (referred to as a 'datapoint') on the map, with a radius of 20m. Each circle could include several buildings, which were identified through satellite photos and numbered as found. One building within the circle was randomly chosen. If it was not a residential building, another building was randomly selected from within the circle. If a circle contained no buildings, sampling was continued by moving on to the next GPS location. If there were just commercial buildings and no residential buildings in that circle, sampling was continued by moving onto the next GPS location. Since the nature of the building could not necessarily be determined from the photo, interviewers went to the location and checked whether or not the chosen building was residential. In either case, whether the circle had no buildings or had no residential buildings, it was included as part of the town area sampled. Since any given residential building could consist of multiple households, pre-prepared random number tables were used to randomly select one household from that building. If it was a one-household building, then that household was automatically chosen for sampling. One adult (aged 18 years or older) was then selected from each household, based on who had the most recent birthday, and then interviewed. This was not truly random sampling, but the bias was considered to be negligible. Each household was attempted to be interviewed up to four times if the designated respondent was not available. If the household was found to be currently unoccupied or abandoned, the next closest home was visited to ask if they knew why that household was empty and whether these neighbours had the contact information for members of the sampled household. As expected, sometimes no information was collected from a specific household if the residents declined to be interviewed or if they had moved out.

---

[1] Of the 53 towns sampled, some were really villages, but I use the term 'towns' to include 'villages' also.

C. Calculation of Probabilities used in the South Lebanon Study[2]

- p_1 = probability of sampling a specific town using PPES
- p_2 = probability of selecting the area within the town (i.e. total area of the circles as a fraction of the town area)[3]
- p_3 = probability of sampling a building in a specific circle (inverse of the total number of buildings in that circle)
- p_4 = probability of sampling a household in the building (inverse of the total number of households in that building)
- p_5 = probability of sampling an adult (18+) in the household (inverse of the total number of adults in that household)[4]

Since $p_1, p_2, \ldots, p_5$ represent independent events, the probability of sampling an individual (p_I) is simply the product of the probabilities,
i.e. $p_I = p_1 * p_2 * p_3 * p_4 * p_5$

Similarly, the probability of sampling a household (p_H) = $p_1 * p_2 * p_3 * p_4$

D. Definition of Weights used in the South Lebanon Study

- For items questioning the individual, the sampling weight was: $\frac{1}{p_I}$
- For items questioning the household, the sampling weight was: $\frac{1}{p_H}$

E. Inherent Issue: No Fixed Probability of Sampling any Given Unit

In this new sampling method, an issue arises regarding the "probability of sampling a particular household in a specific circle"p_3. The issue is that the probability p_3 is not fixed, and may differ depending on which datapoint the household is chosen from and how many households lie in each datapoint circle.

---

[2] Note that the calculations of probabilities and weights discussed in the simulations were different than those used in the South Lebanon study. They have been described in detail in the Methods section.

[3] p_2 is calculated as:
$$\frac{\text{Area of circles sampled in the town (including circles with only commercial buildings or no buildings)}}{\text{Total town area}}$$

[4] This might be labelled as a probability but strictly speaking, only the adult with the most recent birthday is interviewed, so $p_5 = 0$ or 1, depending on whether that adult gives the interview or not.

A given household could fall in the 20m radius circle of two (or more) different datapoints. Each datapoint may have a different number of households in their 20m radius circle. Therefore, the probability of choosing the given household may not be fixed depending on which datapoint's circle the household is selected from. As can be seen from Figure 1.1, a household with coordinates (x,y) falls in datapoint circle 1 (DP 1) and in datapoint circle 2 (DP 2).

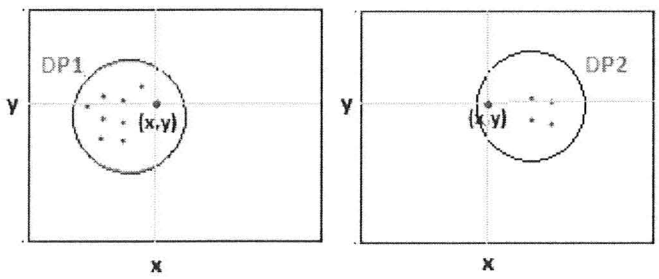


Figure 1.1: A household falling in two datapoint circles, DP 1 and DP 2.

Although the household at (x,y) lies in both circles, the difference in probabilities is due to which datapoint is randomly selected by the researchers to sample the household from, and how many households lie within 20m of that datapoint:

For datapoint 1 – there are 9 households in the circle
For datapoint 2 – there are 5 households in the circle

Accordingly, there are two different probabilities for selecting the household:

For datapoint 1 - P(Selecting the household) = $\frac{1}{9}$
For datapoint 2 - P(Selecting the household) = $\frac{1}{5}$

In other instances, this method could select many datapoints that contain a given household (instead of just two as mentioned in the above example) and sampling without replacement makes it very complicated to determine a single probability of selection for that household. However, a possible way of computing the probability is outlined in Appendix A.

Thus, given that the probability of selecting a household is not fixed, the following questions are explored in this thesis via the use of simulations:

- **Does this affect the estimates of the means or and proportion?**
- **Does this affect the estimates of the standard errors?**

F. Simplifications used in Simulations

This new sampling approach does not produce a fixed a priori probability of sampling a particular household. This issue is explored through simulations run on two hypothetical towns, to understand the effect of having variable a priori probabilities of selection for households in the sample. However, since the original South Lebanon data used in the actual sampling was rather complex, I created simplified versions of the town, and ran some simulations to test the issue of p_3 not being fixed. These simplifications included:

- Using a single town for each of the two simulations (versus multiple towns as were observed in the actual study) and sampling from the population of each single town.
- Assuming each building was a single-household residence, i.e. 1 building = 1 household (hence, both terms are used interchangeably throughout the thesis).
- Dividing the town into two distinct socio-economic status (SES) areas, i.e. high SES area and low SES area.
- Assuming that a response is obtained from each individual interviewed i.e. 100% response rate.

G. Actual Data – Population & Sample Size

In the South Lebanon study, 1,400 households were sampled from multiple towns. The population from which the sample was drawn included 144 towns with 312,209 people and 3 other bigger towns -Tyre, Marjayoun, and Bint Jbeil. Further details of how the actual sample was chosen in the study are provided in Appendix B.

H. Actual Data – Income

There was no perfect way to describe the distribution of actual household income in both the high and low SES areas. In practice, the distribution typically showed a large block in the "middle" with a modest amount of variability, some poorer families, and some with much higher income. This, very roughly, corresponds to a log normal distribution.

"Income" is a continuous variable and will be used as an example in this thesis. It is expected that most people earn a modest income but there are a few who are much better off, thus earning a relatively higher income. Its distribution is roughly a log normal distribution, which is skewed to the right and non-negative. Thus, log of income follows a normal distribution: ***log income*** $\rightarrow$ ***Normal***(μ, σ^2)

I. Actual Data - Proportions of Characteristics (for all households across all towns)

In the high and low SES areas, certain characteristics pertain to some households e.g. number of individuals owning a gun, car, computer etc.

"Proportions of characteristics" are categorical variables with a binary output (i.e. "yes' – characteristic present or "no" – characteristic absent). These variables follow a hyper geometric distribution, more specifically the binomial distribution. If households are sampled completely randomly, which was done using the GPS location method, it is expected that those characteristics will have a binomial distribution. Between the different socio economic status areas, the probabilities differ for various characteristics.

While there were various binary variables present in each of the low and high SES areas, I have just used gun ownership as a generic binary variable in this thesis, for simplicity sake.

2. SAMPLING THEORY

In this section, the theory used for calculating point estimates and standard errors is addressed. When a sample of a specified number of households (HHs) is obtained, there are a number of factors to consider, including the weights and probabilities of selection associated with each household. In addition, any assumptions used when collecting the data or analysing it have to be identified. This information is then used to calculate the point estimates and standard errors.

A. Probabilities

When determining the probability of selecting a household in a given circle, the specific number of households within each circle has to be considered. In the simulations, sampling has been done without replacement so a household cannot be re-sampled if it has already been sampled before. Accordingly, one household represents only one set of households, which lie in the circle from which the sampled household is selected. When sampling without replacement, the probability of selecting a household in each circle depends on the total number of households in the circle which have not already been chosen through another datapoint. In other words, as soon as a household is chosen in the sample, it is treated as if it does not exist anymore, hence cannot be re-sampled. This is in line with what happens in practice, as one would expect to sample a given household only once. In the case that a household is chosen again, one would ignore it and pick a different household, which is effectively sampling without replacement.

Let,

# of HHs not selected in circle = # of HHs in circle – # of HHs already selected in circle

Then,

$$p_H = \frac{1}{\text{\# of HHs not selected in circle}}, \qquad \text{if \# of HHs not selected in circle} > 0$$

B. Weights

When a given household is sampled from a circle, it "represents" a number of households in the entire population, in order for the whole sample to be representative of the actual population. The weight of a selected household is just the reciprocal of the probability of selection, so weight is defined as:

$$W_H = \frac{1}{\frac{1}{\text{\# of HHs not selected in circle}}}$$

= # of HHs not selected in circle
= # of HHs in circle - # of HHs already selected in circle

Adjusted Weight

When obtaining the weighted calculations, adjusted weight (w'_j) was used, where $w'_j = w_j/\overline{w}$. Since the denominator for standard deviation and standard error calculations includes the term $\sum_{j=1}^{n} w_j$, by replacing w_j with w'_j, the denominator $\sum_{j=1}^{n} w'_j = n$. This ensures that the sum of weights is not exaggerated in the calculations and always equals n, regardless of the size of the population from which the sample is drawn. Using weights w_j and adjusted weights w'_j, weighted calculations are obtained for sample parameters.

C. Population Parameters – Mean/Proportion & Standard Deviation (SD)

The population parameters, mean/proportion and standard deviations, for income and gun ownership are defined below:

Income:

Given x_i = income per household for each household i in the population,

Population Mean: $\mu = \frac{1}{N}\sum_{i=1}^{N} x_i$,

Population SD: $\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}$

Gun Ownership:

Given x_i = 0 or 1 for each household i in the population[5],

Population Proportion: $p = \frac{1}{N}\sum_{i=1}^{N} x_i$

Population SD: $\sigma = \sqrt{p(1-p)}$

[5] Gun owned: 0 = No, 1 = Yes

D. Sample Parameters – Mean/Proportion & Standard Deviation (SD)[6]

All estimates calculated in the sample were weighted, for a sample of n observations. The variables p_j, w_j, and w'_j were used, defined as:

p_j = probability of choosing household j

$w_j = \frac{1}{p_j}$, **and** $w'_j = \frac{w_j}{\bar{w}}$, $\quad j = 1, 100$

Income:

Given x_j = income per household for each household j in the sample,

Weighted Sample Mean: $\bar{x} = \frac{\sum_{j=1}^{n} w'_j x_j}{\sum_{j=1}^{n} w'_j}$

Weighted Sample SD: $s_{\bar{x}} = \sqrt{\frac{\sum_{j=1}^{n}[w'_j(x_j - \bar{x})^2]}{\left[\sum_{j=1}^{n} w'_j\right] - 1}} = \sqrt{\frac{\sum_{j=1}^{n}[w'_j(x_j - \bar{x})^2]}{n-1}}$

Gun Ownership:

Given $x_j = 0$ or 1 for each household j in the population[7],

Weighted Sample Proportion: $\bar{p} = \frac{\sum_{j=1}^{n} w'_j x_j}{\sum_{j=1}^{n} w'_j}$,

Weighted Sample SD: $s_{\bar{p}} = \sqrt{\frac{\sum_{j=1}^{n}[w'_j(x_j - \bar{p})^2]}{\left[\sum_{j=1}^{n} w'_j\right] - 1}} = \sqrt{\frac{\sum_{j=1}^{n}[w'_j(x_j - \bar{p})^2]}{n-1}}$

In both calculations of weighted sample standard deviation above, in the denominator $\left[\sum_{j=1}^{n} w'_j\right] = n$. This is because:

$$\begin{aligned}\sum_{j=1}^{n} w'_j &= \sum_{j=1}^{n} w_j/\bar{w} \\ &= \frac{1}{\bar{w}} * \sum_{j=1}^{n} w_j \\ &= \frac{1}{\sum_{j=1}^{n} w_j/n} * \sum_{j=1}^{n} w_j \\ &= \frac{n}{\sum_{j=1}^{n} w_j} * \sum_{j=1}^{n} w_j = n\end{aligned}$$

---

6 (Madansky n.d.)

7 Gun owned: 0 = No, 1 = Yes

E. Sample Parameters –Standard Error (SE)

In both calculations of the weighted sample standard errors below, the standard error has been multiplied by the finite population correction (FPC) = $\sqrt{1-f}$, because a substantial proportion of the population was sampled in the simulations[8].

Income:

General Weighted Sample SE: $SE_{\bar{x}} = \frac{\text{weighted standard deviation}}{\sqrt{\sum_{j=1}^{n} w'_j}} = \frac{s_{\bar{x}}}{\sqrt{n}}$

Weighted Sample SE after incorporating the FPC = $\frac{s_{\bar{x}}}{\sqrt{n}} * \sqrt{1-f}$

Gun Ownership:

General Weighted Sample SE: $SE_{\bar{p}} = \frac{\text{weighted standard deviation}}{\sqrt{\sum_{j=1}^{n} w'_j - 1}} = \frac{s_{\bar{p}}}{\sqrt{n-1}}$

Weighted Sample SE after incorporating the FPC = $\frac{s_{\bar{p}}}{\sqrt{n-1}} * \sqrt{1-f}$

---

8 f = sampling fraction = $\frac{sample\ size}{population\ size} = \frac{n}{N}$

3. METHODS

In this section I will describe the simulations, the parameters required for creating the town and how the simplified towns were set up. Included in this section are the distributions for income and gun ownership properties associated with each household. I will also describe how the samples were drawn from the population, and the calculation of probabilities and weights for each household in the sample.

A. Parameters

Various parameters were used to create each simulated town, from which a large number of samples were then drawn. The parameters were kept completely general in the code so they could be changed for any run of the code to create different layouts of the town.

Table 3.1 shows a list of the parameters used in constructing the town, along with a brief description of each parameter. All eight parameters were specified for each socio-economic status (SES) area.

Parameter Name	**Parameter Description**
Number of Households	Number of households in the town
Minimum Distance between Households	Minimum distance between the centres of each household
Minimum Distance from Boundary	Minimum distance required between each household and the town's edge when placing households in the town
Town Size (L x W)	Dimensions of the town, where L = length and W = width
Distance from West Boundary	Boundary line that divides high and low SES areas in the town
Proportion with Characteristic (e.g. Gun Ownership)	Proportion of population having a particular characteristic (applicable to any binary variable)
Mean of Income (units)	Proposed mean income of each SES area

Variance of Income (units)	Proposed variance of log income of each SES area

Table 3.1: List of parameters for creating a simulated town.

In the simulations run for the purposes of this thesis, only the town size was varied. However, in principle all these parameters could be varied to create numerous hypothetical towns.

In order to draw a sample, some additional parameters had to be specified. Again, the parameters were kept completely general in the code so they could be changed for any run of the code to create different samples. Table 3.2 shows the additional two parameters required in creating a sample, along with a brief description of each parameter.

Parameter Name	**Parameter Description**
Sample Size	Required size of the sample (n)
Datapoint Radius	Radius of the circle drawn around each datapoint to sample from (r)

Table 3.2: List of parameters for drawing a sample from the simulated town.

B. Counts

Table 3.3 shows a list of six counts that were also used to keep track of information in the sample. Using the counts, the percentages of the datapoints and households sampled from the high and the low SES areas respectively, were calculated.

Count Name	**Count Description**
Total Datapoints Sampled	Total number of datapoints sampled to get to the required sample size of n datapoints with at least one household that had not already been sampled before
High SES Count	Number of datapoints sampled from the high SES area
Low SES Count	Number of datapoints sampled from the low SES area

Total Buildings Sampled	Equals n, the sample size (Note that n = 100 throughout the thesis)
High SES Count	Number of households sampled from the high SES area
Low SES Count	Number of households sampled from the low SES area

Table 3.3: List of counts recorded for each sample.

C. Simulations

Simulations were run on two hypothetical towns, to obtain estimates of parameters of interest and standard errors. There were two aspects to the simulation including, creating a town and then sampling from it.

The town was created in Excel using Visual Basic for Applications (VBA), and had the following characteristics:

Town Size –

Town size = L x W units, where L and W were the length and width of the town, respectively.

Coordinates –

North(N) / South(S) coordinates: Min: 0 Max: L units.
East(E) / West(W) coordinates: Min: 0, Max: W units.

SES Areas –

The town was divided into two areas, one more affluent than the other. The high SES area had N_h households and the low SES area had N_l households.

West Boundary –

The west boundary divided the high and low SES areas, assuming both areas were enclosed within a rectangle. The boundary was set at B units along the East-West axis and calculated in the following way:

1. First F_H was calculated, where F_H = fraction of the high SES area in town.
2. Once F_H was known, the boundary was set at $B = (F_H * W) + 0.5$ units.

1. SIMULATING A STANDARD TOWN:

i. Town Characteristics

In the first simulation, the town characteristics were as follows:

Town size = 1000 * 1000 units, i.e. L = 1000 units and W = 1000 units
Fraction of the high SES area in town = F_H = 20%
Boundary = B = (F_H * W) + 0.5 = (20% * 1000) + 0.5 = 200.5 units
N_l = 900 buildings in the low SES area, N_h = 100 buildings in the high SES area

The minimum distance between buildings was specified to be 5 units in the low SES area and 10 units in the high SES area.

Given the town size, boundary, minimum distance between buildings and the population of each SES area, the density of each area was defined. Figure 3.1 shows the town sketch, for a town of size 1000 x 1000 units:

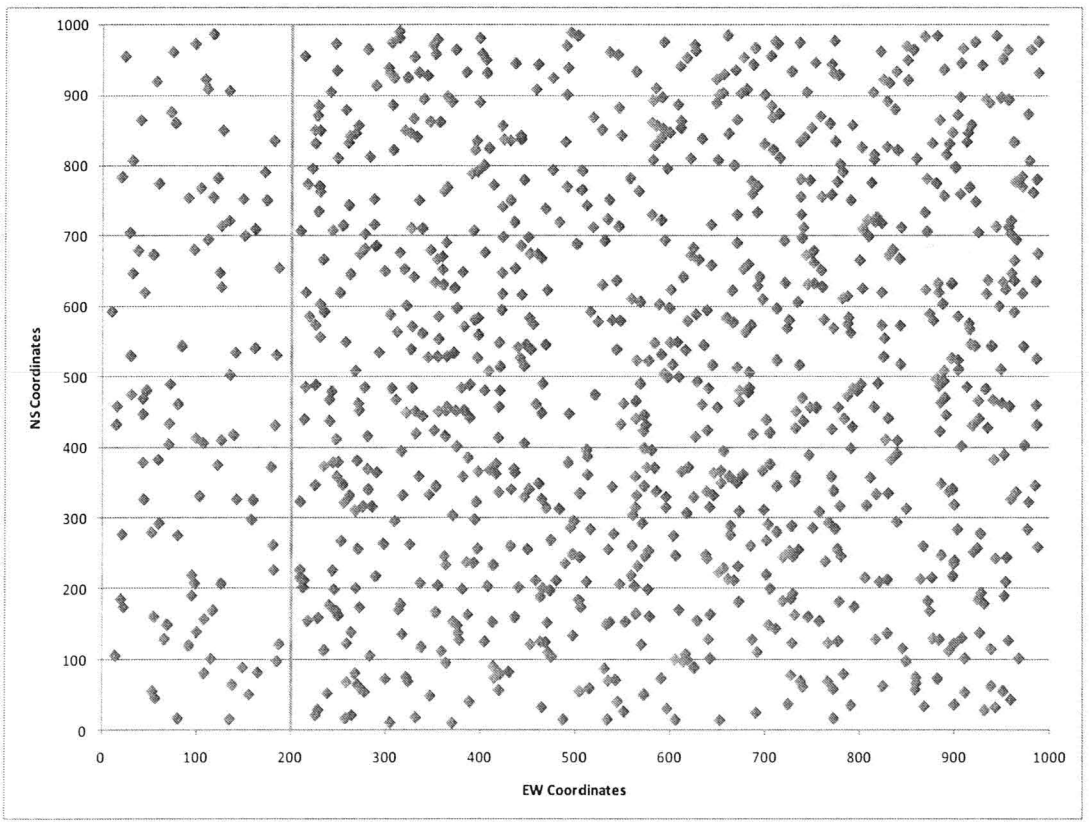


Figure 3.1: Sketch of town of size 1000 x 1000 units with boundary 200.5 units, and 900 buildings in the low SES area and 100 buildings in the high SES area.

ii. Adding Buildings to the Town

N buildings were placed in the town in the following way, where N_h buildings were from the high SES area and N_l buildings were from the low SES area. It was assumed that there was only one household per building. The following steps were used:

1. Generated EW/NS coordinates for the centre of each building using a Random Number Generator (RNG). The random number was scaled to the appropriate range for both sets of coordinates (i.e. 0 to 1000).

   For the high SES area (N_h buildings):
   - NS - Randomly sampled a number between 0 and L e.g. 147
   - EW - Randomly sampled a number between 0and B e.g. 25

   For the low SES area (N_l buildings):
   - NS - Randomly sampled a number between 0 and L e.g. 400
   - EW - Randomly sampled a number between B and W e.g. 876
   - EW coordinates of 0 - B covered the more affluent part of town
   - EW coordinates B - W covered the less affluent part of town

2. Each new set of coordinates was compared against the set of coordinates of all household centres already added in the town. It was ensured that all centres were a minimum distance apart. If a centre violated the minimum distance condition, another household centre was created using a RIG.

Table 3.4 shows a partial list of buildings in the low and high SES area, along with their respective centre coordinates. In the simulation, the low SES area was made very dense with 900 buildings in it, whereas the high SES area was much more spaced out with just 100 buildings.

Low SES Area		
Building Number	**X-Coordinate**	**Y-Coordinate**
1	974	104
2	505	539
3	596	391
--	--	--
898	819	663
899	982	314
900	733	232

High SES Area		
Building Number	**X-Coordinate**	**Y-Coordinate**
901	9	170
902	114	252
903	107	909
--	--	--
998	167	326
999	46	550
1000	96	487

Table 3.4: Partial list of buildings in the town with their coordinates.

iii. Distributions for Income and Characteristics

1. Income:

In practice, the distribution typically showed a large block in the "middle" with a modest amount of variability, some poorer families, and some with much higher income. Thus, as noted in the earlier sections, a lognormal distribution has been proposed for "income" in each of the high and low SES areas. Figures 3.2 and 3.3 below show graphs of the expected/theoretical distributions (the distributions for the simulated town are shown in the Methods section later). As shown in the graphs (and also in reality), there is likely to be some overlap between both distributions, implying that there are some high income earners who live in the low SES area, or some low income earners who live in the high SES area.

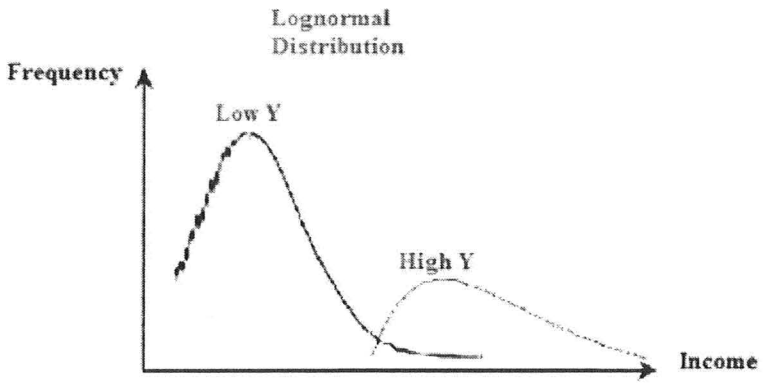


Figure 3.2: Income ~ Log Normal(μ, σ^2)

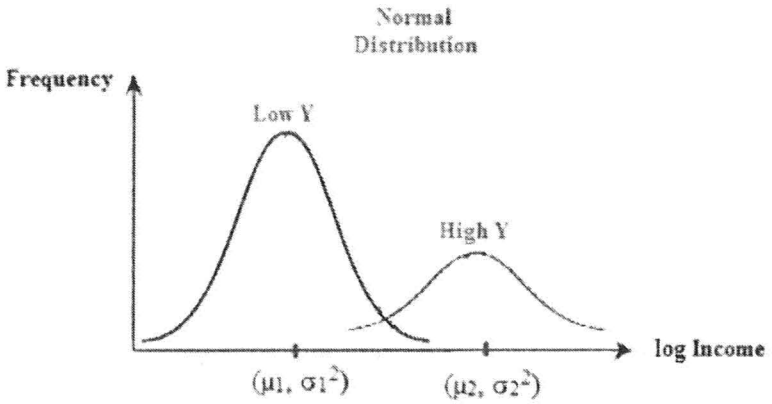


Figure 3.3: Log Income ~ Normal(μ, σ^2)

For the simulation, it was assumed that the distribution of income corresponding to the high SES area would have a higher mean, since those with higher SES are generally wealthier. Thus, the following log-normal distributions were proposed for household income of both areas:

Let,

i. Household income for low SES area ~ Log Normal(μ_l, σ_l^2), **and** $\mu_l = 6$, $\sigma_l^2 = 1$
ii. Household income for high SES area ~ Log Normal(μ_2, σ_2^2), **and** $\mu_2 = 10$, $\sigma_2^2 = 1.5$

Income was then allocated to all N households in the simulated town as follows:

Choose,

i. N_l random numbers (R#'s) from a N(6, 1)
ii. N_h random numbers (R#'s) from a N((10, 1.5)

Since log income was normally distributed, then income = $e^{R\#}$

2. Characteristics:

There were many characteristics that were observed in the actual South Lebanon study, e.g. gun ownership, car ownership etc. The probabilities for households having specific characteristics differed between the high and low SES areas, for various reasons.

In the simulation, only gun ownership was considered for simplicity's sake. A higher gun ownership probability was proposed for households in the low SES area, and a lower one for households in the high SES area. These probabilities were then allocated to randomly selected households in the simulated town. Gun ownership is an example of a binary variable, with probability of household owning a gun π. Thus,

Gun ownership ~ Binomial (N_i, π_i)

where,

i = l or h
N= Number of households in each of the high and low SES areas
- N_l = number of households in the low SES area
- N_h= number of households in the high SES area

π = Gun ownership proportion in the town
- π_l = proportion of households with gun ownership in the low SES area
- π_h = proportion of households gun ownership in the high SES area

Note that in this thesis, only proportions of characteristics (namely gun ownership) have been looked at. However, it is possible to estimate number of guns. More details are given in Appendix C.

iv. Spreadsheet of Buildings in Town

Table 3.5 shows part of the spreadsheet that was kept for storing information of all N buildings in the town. For each building, the building number, centre coordinates, income earned and gun ownership properties were noted. Since gun ownership was treated as a binary variable, 0 meant no gun owned, and 1 meant gun owned. The partial list of buildings for the N_l buildings in the low SES area and N_h buildings in the high SES area, is as below:

Low SES Area				
Building Number	**X-Coordinate**	**Y-Coordinate**	**Income**	**Gun Ownership**
1	974	104	190	0
2	505	539	711	1
3	596	391	275	0
--	--	--	--	--
--	--	--	--	--
898	819	663	186	0
899	982	314	452	1
900	733	232	271	0

High SES Area				
Building Number	**X-Coordinate**	**Y-Coordinate**	**Income**	**Gun Ownership**
901	9	170	6,371	0
902	114	252	2,961	0
903	107	909	18,921	0
--	--	--	--	--
--	--	--	--	--
998	167	326	81,190	1
999	46	550	8,780	0
1000	96	487	27,672	0

Table 3.5: Partial list of buildings in the town with their coordinates, income earned and gun ownership properties[9].

[9] Gun ownership: 1 = Yes, 0 = No

In the simulation, households with the gun ownership characteristic were coloured in red. The low SES area has $\pi_l = 10\%$ and the high SES area has $\pi_h = 7\%$. So there are 90 low SES households[10] that owned guns and 7 high SES households[11] that owned guns. Figure 3.4 shows the town sketch, with 97 households[12] out of 1000 households owning a gun (drawn as circles on the plot).

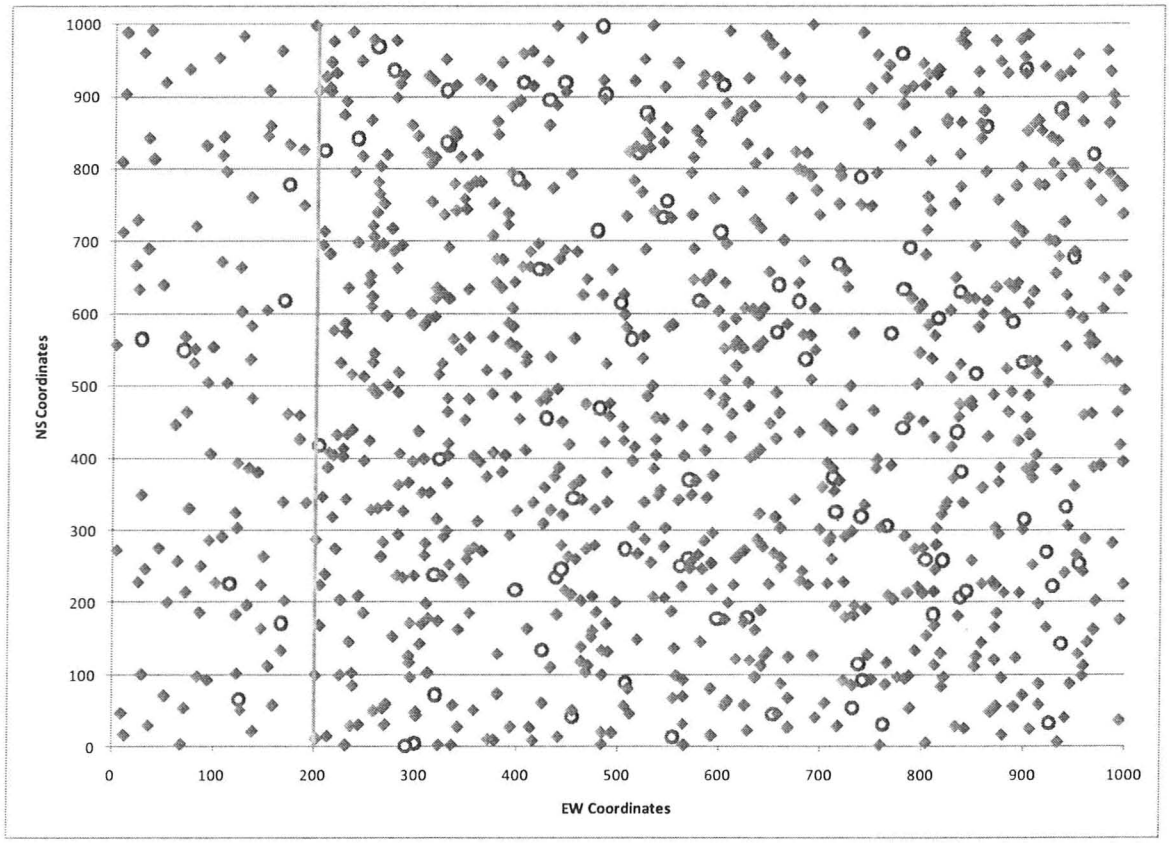


Figure 3.4: Sketch of town containing 1000 households (drawn as diamonds on the plot), out of which 97 households own a gun.

v. Drawing a Sample of n Households

Figure 3.5 shows the town with N households from which the sample of n households was drawn. The high SES area is between the EW coordinates 0 and B, and the low SES area is between the EW coordinates B and W.

10 $\pi_l * N_l = 10\% * 900$ households = 90 households owning a gun

11 $\pi_h * N_h = 7\% * 100$ households = 7 households owning a gun

12 90 + 7 = 97 households owning a gun in the whole town

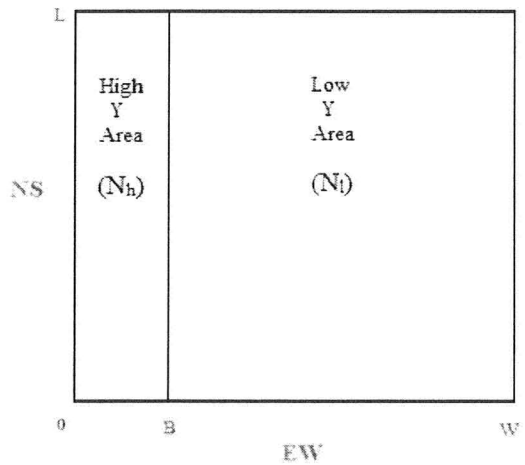


Figure 3.5: Sketch of town from which the sample is drawn.

These steps were repeated for each randomly selected datapoint in the sample:

1. Randomly selected datapoint locations to sample, by randomly choosing EW and NS coordinates for the datapoint.

2. Created a radius, r, around each chosen datapoint. Counted all the relevant buildings that fell within the radius circle of the datapoint. Figure 3.6 shows that if the DP centre was (x,y), then any household within r units of its radius, including a household inside the circle or on the circle's edge [e.g. (x_1, y_1)], would be a relevant household, satisfying: $\sqrt{(x - x_1)^2 + (y - y_1)^2} \leq$ r units

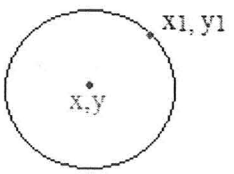


Figure 3.6: A household on the edge of the circle is included for sampling purposes.

3. Identified whether the datapoint was in the high or low SES area, by looking at the x-coordinate of each datapoint. If the x-coordinate $\leq W_h$, the datapoint fell in the high SES area, otherwise it was in the low SES area. This was done to check if the proportion of datapoints sampled from the high and low SES areas matched what was expected).

4. If the datapoint circle contained buildings that had not already been sampled, one of those buildings was randomly sampled. For that building, information such as its building number, centre coordinates, income earned and gun ownership properties was stored.

5. Calculated the probabilities and weights associated with that building.

6. Also, the program outputted information of other circles that overlapped with that specific datapoint. This gave an idea of how often the sampled datapoints overlapped with each other.

The goal was to collect information for a sample size of n buildings. In other words, n datapoints had to be sampled with each datapoint containing at least one household that had not already been sampled. However, sometimes a datapoint was randomly selected containing no buildings or buildings that had already been sampled. In such a case, that datapoint was counted as part of the total number of datapoints sampled to get to the final sample of n buildings. But that datapoint was not added to the required sample of n datapoints since no households from its radius circle could be sampled. Once the sample of n households was collected, the point estimates for income and gun ownership were calculated.

vi. Weights and Probabilities for Five Scenarios

In reality, buildings can be built on the boundary. Accordingly, in the simulation the buildings could be placed on the boundary to imitate reality by setting the "minimum distance from the boundary" parameter to 0.

Figure 3.7 shows that households close to the boundary were less likely to be chosen in the sample. Looking at HH1, the shaded region outside the town's edge included all those datapoints which were within r units of the household, but could not be used to sample HH1 with since they fell outside the town. However, HH2 could be sampled using all datapoints which were within r units of the household. Thus, HH1 could be sampled with fewer datapoints compared to HH2, since HH1 was close to the boundary. Similarly, HH3 and HH4 could only be sampled with fewer datapoints compared to HH2, since they too were close to the boundary.

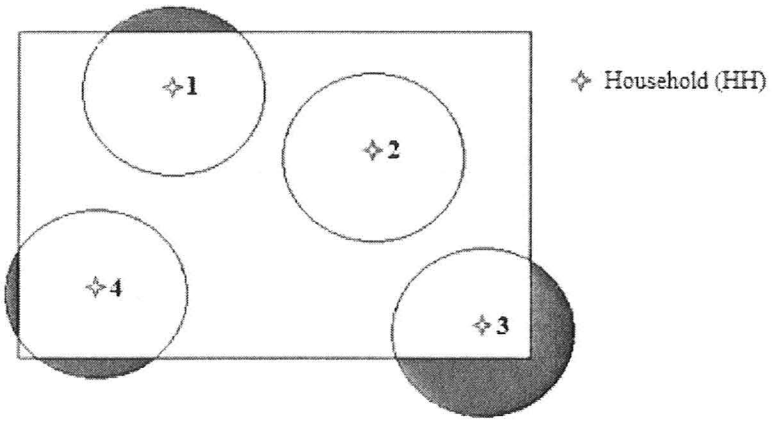


Figure 3.7: Examples of households far from and close to the town's boundary[13].

In the simulation, the probability of choosing a household from a selected datapoint circle was calculated, using one of five different approaches that are defined below. Once the probability was obtained, the weight was also calculated as follows:

1. First, it was determined which scenario the datapoint fell under.
2. Then, the corresponding approach was used to calculate the probability of selecting a HH from that datapoint.
3. Finally, weight of that HH was calculated (reciprocal of the probability).

Defining Variables for the 5 Scenarios:

In order to determine which scenario the datapoint fell in, a few variables had to be defined. These included:

T = Area of whole town in which the buildings lie = town length * town width
h = number of households in a datapoint circle
r = radius of circle
k = x-coordinate distance between the datapoint and town boundary
= min(x coordinate of datapoint, town length – x coordinate of datapoint)
j = y- coordinate distance between the datapoint and town boundary
= min(y coordinate of datapoint, town width – y coordinate of datapoint)
A = unshaded area of the circle, which is inside the town

So for each of the five scenarios illustrated in Figures 3.8a - 3.8e below, the respective conditions and weight and probability expressions are as defined:

[13] HH 1, 3 and 4 were close to the boundary, hence had a lower probability of being chosen. HH 2 was within the town, thus had a higher probability of being chosen.

1. If k and j ≥ r from boundary, then the circle area of interest = $(r^2)\pi$

Hence, P(choosing a household) = $\frac{r^2\pi}{T} * \frac{1}{h}$

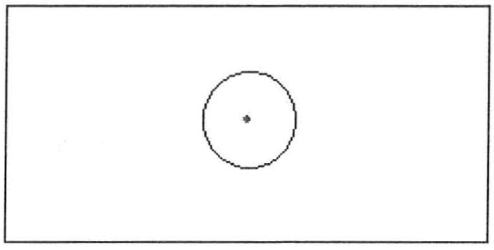


Figure 3.8a: Scenario 1 – Complete circle is within the town.

2. If j ≥ r and k < r from boundary, then circle area of interest = A

Hence, P(choosing a household) = $\frac{A}{T} * \frac{1}{h}$

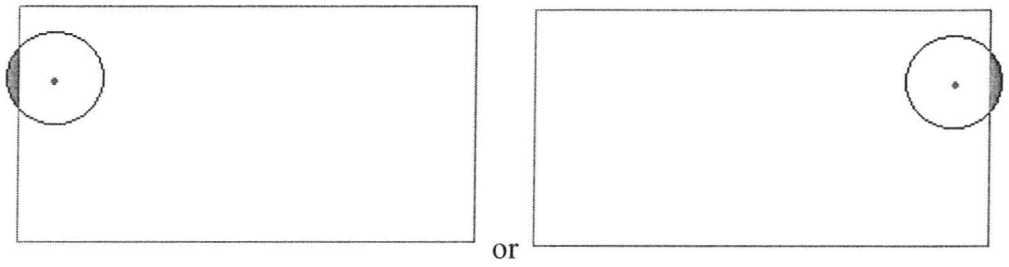


Figure 3.8b: Scenario 2 – Part of circle is outside the town on the x-axis.

3. If k ≥ r and j < r from boundary, then circle area of interest = A

Hence, P(choosing a household) = $\frac{A}{T} * \frac{1}{h}$

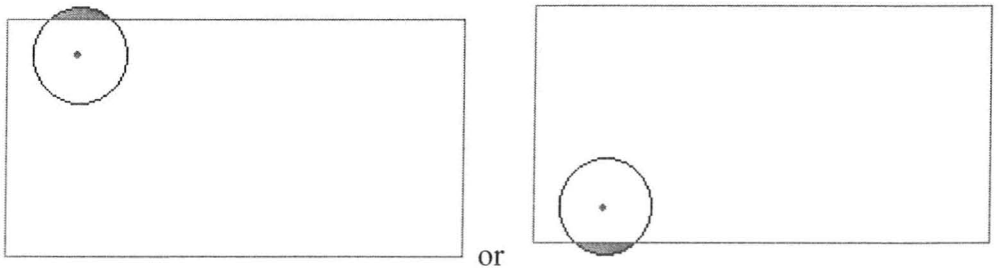


Figure 3.8c: Scenario 3 – Part of circle is outside the town on the y-axis.

4. If $j^2 + k^2 < r^2$ (i.e. j < r and k < r), then circle area of interest = A

Hence, P(choosing a household) = $\frac{A}{T} * \frac{1}{h}$

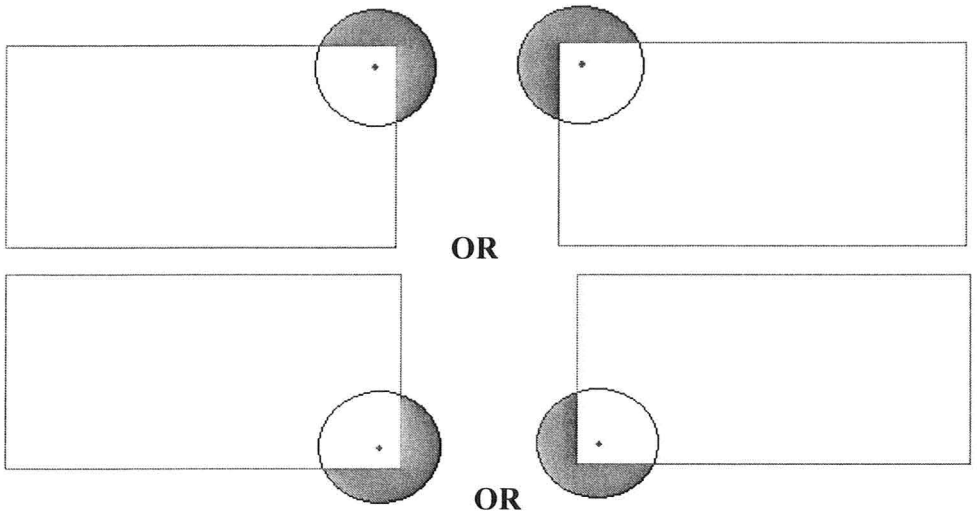


Figure 3.8d: Scenario 4 – Examples show situations at each corner; circle includes the corner but part of circle is outside the town.

5. If $j^2 + k^2 > r^2$ and j < r and k < r, then circle area of interest = A

Hence, P(choosing a household) = $\frac{A}{T} * \frac{1}{h}$

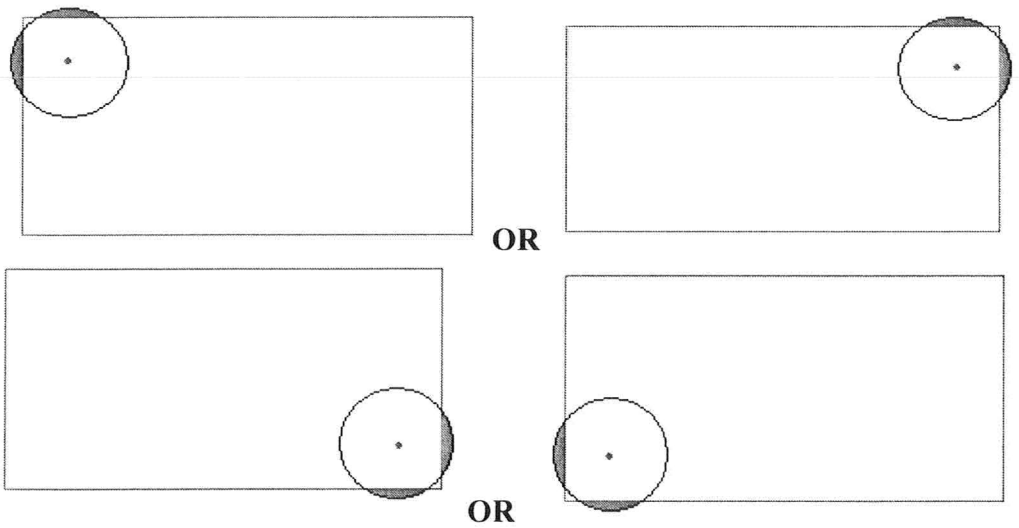


Figure 3.8e: Scenario 5 – Examples show situations at each corner; part of circle is outside the town on both axes but corners are outside the circles.

The minimum an unshaded area can be is $\frac{1}{4}\pi r^2$, when a datapoint lies on one of the four corners of the town. The maximum an unshaded area can be is πr^2, when the circle around the datapoint is completely inside the town.

Once it was determined which scenario a datapoint fell under, the probability of choosing a household (P) and its weight were both calculated. The variable "# of households not selected in the radius" was used, which was the difference between "# of households in the circle (h)" and "# of households already selected in the circle". As a result, the following calculations for probabilities and weights were obtained, assuming radius r = 20 units.

Let,

HHNS = number of households not selected in the circle

1. If a DP centre is $\geq$ 20units from the boundary, then:

$$\text{P(choosing a household)} = \frac{400\pi}{T} * \frac{1}{\text{HHNS}}$$

$$\text{Weight(of sampled household)} = \frac{1}{P} = \frac{\text{T}}{400\pi} * \text{HHNS}$$

2. If a DP centre is $<$ 20units from the boundary, then:

$$\text{P(choosing a household)} = \frac{\text{A}}{T} * \frac{1}{\text{HHNS}}$$

$$\text{Weight(of a sampled household)} = \frac{1}{P} = \frac{\text{T}}{\text{A}} * \text{HHNS}$$

By accounting for boundaries and households near the boundaries, the proper probabilities and weights for sampling a household were obtained. The full details for calculating the area of the circle within the town, under the five different approaches, is in Appendix D.

vii. Test of Overlap

Given the random selection of datapoints when drawing a sample from the population, there were likely to be situations where some circles overlapped with other circles (with radius r units). In order to see how often that happened and which circles overlapped with other circles, a test was run for each datapoint in the sample. In the test, each datapoint was compared against all other datapoints in the sample to see if the overlap criterion was met.

Figure 3.9 shows a situation in which two circles do not overlap each other, since the distance between them is equal to more than twice the radius (r).

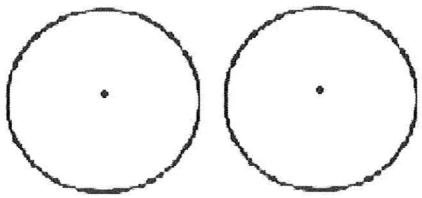
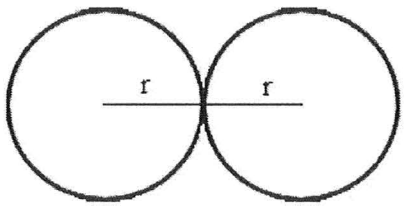
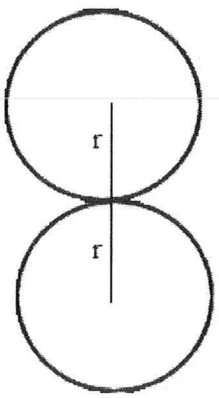


Figure 3.9: A case of two non-overlapping circles surrounding two datapoints.

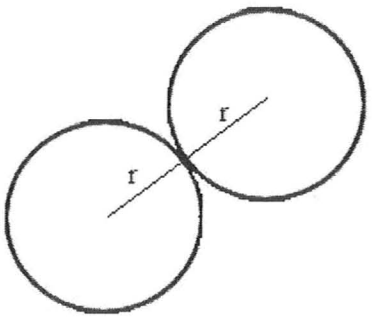
Figure 3.10 shows three more situations in which the two circles do not overlap each other, since the distance between them is equal to twice the size of the radius



Case A



Case B



Case C

Figure 3.10: Three cases of two non-overlapping touching circles, with a faint possibility that a HH is located at the point of contact between both circles.

These are, though, touching circles and there is a faint possibility that a HH is located at the point of contact between both circles. In case A the y coordinates of both centres are equal, in case b the x coordinates of both centres are equal, and in case C the coordinates of both centres differ by exactly 32 and 24 units on the x and y axis (so that the centres of the circles are exactly 40 units apart). However, touching circles have been ignored in this thesis.

Figure 3.11 shows a situation in which two circles were overlapping each other, given the datapoint of one circle was (x_1, y_1) and (x_2, y_2) for the other. In other words, the distance between both datapoints was less than twice the radius.

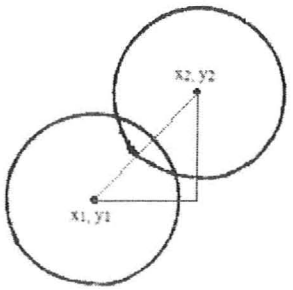


Figure 3.11: A case of two overlapping circles surrounding two datapoints.

In the simulation, to determine if circles overlapped each other in the sample, this condition had to be met for each circle: $\sqrt{(\mathbf{x_2 - x_1})^2 + (\mathbf{y_2 - y_1})^2} \leq \mathbf{2r}$ **units**

If the above was true for the circle in question, then it overlapped the other circle. This observation was noted for each circle in the whole sample.

viii. Obtaining Multiple Samples of a Specific Size Sample

A macro was created that produced multiple samples (iterations) for a given sample size. Each sample outputted calculations for the income and gun ownership weighted mean and standard error calculations. Those four calculations, for each sample, were recorded on another worksheet. Accordingly, further analysis was then done using the results compiled for each set of iterations on the simulated town.

2. SIMULATING A DENSE TOWN:

In the first simulation, samples were drawn from a standard town, which had dimensions of 1000 x 1000 units. In the second simulation, samples were drawn from a dense town. All variable values were kept the same as in the standard town, except the town size, which was shrunk to a quarter of the size of the standard town. Thus, the dimensions of the dense town were 500 * 500 units.

4. EXPECTATIONS FROM THEORY

In this section, I will discuss what I expect from theory to see whether it compares to the results obtained from the two simulations.

A. Calculations

Two simulations were conducted on two different towns – a standard and a dense town. All variables were kept the same between both towns, e .g. number of buildings etc, except town size, which was varied. The dimensions for the two towns were as follows:

Standard – The dimensions of the town size were 1000 * 1000 units.
Dense – The dimensions of the town size were 500 * 500 units.

Figure 4.1 shows the sketch of both towns from which the samples were drawn.

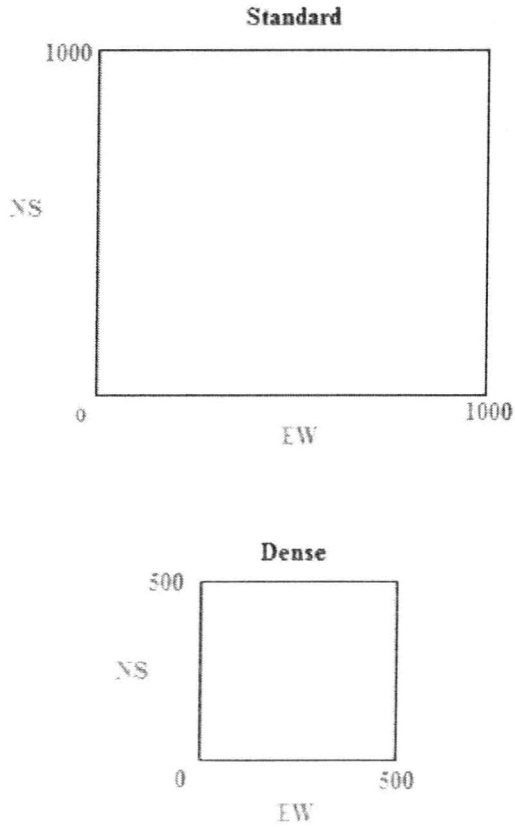


Figure 4.1: Sketch of the standard town and the dense town.

A sample size of 100 households was obtained from the population of 1000 households, for each town. Many additional samples (iterations) chosen in the same way were obtained. For each iteration, the weighted average (or proportion) and standard error of income and gun ownership were calculated.

Variable j -

In the simulation, a single sample drawn from the population consisted of n datapoints with at least one household that had not already been sampled before. Since sampling was being done without replacement, if a given household was chosen a second time (or more), it would be discarded and another one would be randomly sampled. In the calculations of the mean, standard deviation and standard error, the variable *j* looped from 1 to n, to keep count of the n datapoints.

Weighted Mean -

$$\bar{x}_w = \frac{\sum_{j=1}^{n} w_j x_j}{\sum_{j=1}^{n} w_j}$$

When calculating the mean, weights had to be accounted for, thus resulting in weighted means. The calculation for weights is as defined under the Methods section. Once the weight (w_j) was calculated for each household, adjusted weight w'_j was then calculated for each household (where $w'_j = \frac{w_j}{\bar{w}}$). Accordingly, this allowed $\sum_{j=1}^{n} w'_j = n$, for the standard deviation and standard error calculations.

Weighted Standard Deviation (SD) -

$$s_w = \sqrt{\frac{\sum_{j=1}^{n} [w'_j (x_j - \bar{x}_w)^2]}{\left[\sum_{j=1}^{n} w'_j\right] - 1}}$$

Weighted Standard Error (SE) of a Sample Mean -

$$SE_w = \frac{\sqrt{\frac{\sum_{j=1}^{n} [w'_j (x_j - \bar{x}_w)^2]}{\left[\sum_{j=1}^{n} w'_j\right] - 1}}}{\left[\sum_{j=1}^{n} w'_j\right]} = \frac{\sqrt{\frac{\sum_{j=1}^{n} [w'_j (x_j - \bar{x}_w)^2]}{n-1}}}{n}$$

Weighted Standard Error (SE) of a Mean of Sample Means -

The Central Limit Theorem (CLT) shows that the distribution of sample means asymptotically approaches a normal distribution as n increases (regardless of the true underlying distribution of the population that the sample is drawn

from).Therefore, the mean of n_s sample means is also asymptotically normally distributed. The weighted standard error of this "mean of means" (or the standard deviation of the distribution of mean of means) is,

$$\frac{Weighted\ Standard\ Error\ (SE) for\ a\ Sample\ Mean}{\sqrt{n_s}}$$

B. Finite Population Corrector (FPC)

When calculating the standard error, the finite population correction (FPC) was also incorporated in the calculation, since the town was of a finite size. The FPC is used to ensure an unbiased estimate of the SE when sampling without replacement, which is how sampling was done in the two simulations in this thesis.

Sampling theory defines the sampling fraction as $f = \frac{sample\ size\ (n)}{population\ size\ (N)}$.

When f is very small (less than 5%), the margin of error for a particular sampling method is nearly the same as when the sample is taken from an infinite population. However, when the fraction is greater than 5%, one can account for the greater precision gained by sampling a non-negligible percentage of the population. This is done by adjusting the margin of error using the finite population correction (FPC). So, as the sample size approaches the population size, the FPC approaches zero, thus eliminating the margin of error. In other words, when you sample close to the entire population, you expect little sampling error.

$$FPC = \sqrt{\frac{N-n}{N-1}} \text{ or } FPC \sim \sqrt{1-f}$$

In the simulation since $f = \frac{100}{1000} = 10\% > 5\%$, the FPC was incorporated in the calculations of standard error for income and gun ownership.

C. Effect on Weights and Probabilities of Increased Density

The standard town is of size A, and the dense town is of size 0.25A, so the density expected of the dense town is four times higher than the density of the standard down. This is illustrated using the simulations discussed in the thesis. With simulation 2, when the town's physical size was decreased, the town becomes denser. Previously, the high SES area had a density of 0.5

households/1000 square units[14] and the low SES area had a density of 1.125 households/ 1000 square units[15]. After decreasing the town size, the high SES area has density of 2 households/1000 square units[16] and the low SES area has density of 4.5 households/1000 square units[17]. So when the town is four times denser compared to the standard town, one expects the weights (w) to also be on average four times higher in the denser town (because any circle in the dense town, compared to a circle in the standard town, will have on average four times as many more households in it, as shown in Figure 4.2 below). It is also expected that more overlapping will occur in the denser town, since the number of circles/unit area is greater in the denser town compared to the standard town.

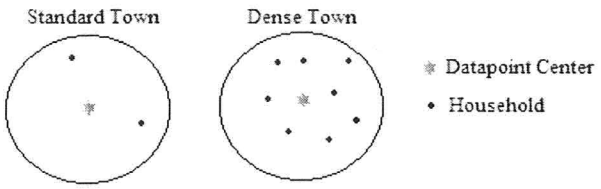


Figure 4.2: Given the dense town is four times denser than the standard town, it is expected that on average the number of households in the dense town will be four times more than the number of households in the standard town.

However, on average the weight may not actually be different between both towns, because overall it is a mixture of the fraction of area circles take up and the density of the town, as illustrated in the example below.

Example:

Assume,

Area of the standard town = A (which is four times the size of the dense town)
Area of the dense town = $\frac{A}{4}$
Area of any selected datapoint circle in either town = πr^2
Number of households in a circle in the standard town = HH_s
Number of households in a circle in the dense town[18] = $4HH_s$

---

[14] $\frac{100\ households}{(200*1000)\ units\ square\ area} * 1000$

[15] $\frac{900\ households}{(800*1000)\ units\ square\ area} * 1000$

[16] $\frac{100\ households}{(100*500)\ units\ square\ area} * 1000$

[17] $\frac{900\ households}{(400*500)\ units\ square\ area} * 1000$

[18] Since the dense town is four times denser than the standard town, it is expected that on average the # of households in the dense town will be four times greater compared to the standard town.

Then,

Proportion of standard town covered by the circle surrounding a selected datapoint = $\frac{\pi r^2}{A}$

Proportion of dense town covered by the circle surrounding a selected datapoint = $= \frac{\pi r^2}{\frac{A}{4}} = \frac{4\pi r^2}{A}$

P(selecting a household from a circle in the standard town) = $\frac{1}{HH_S}$

P(selecting a household from circle in the dense town) = $\frac{1}{4HH_S}$

Now, given the probability of selection for a sampled household is a product of P(selecting a household from circle) and the proportion of town area covered by the selected circle, the probability of selection for a sampled household in both towns is equal[19]. This holds true since the fraction of area and density of town offset each other in the calculation, even though the dense town is four times denser than the standard town. Accordingly, since weight is just a reciprocal of the probability of selection, it also is the same. So on average, weight may not be different, because overall it is a mixture of the fraction of area circles take up and greater density.

D. Probability of Choosing Households in the High & Low SES Areas

When selecting a datapoint, there may be no households (eligible or otherwise) within the radius of that datapoint. The probability of this occurrence is affected by the density of area around that datapoint, where density is the number of households per unit area. In general, it is lower for higher density areas. As illustrated below, this variation in density has an effect on the expected probability of choosing a high/low SES area household.

In the simulated town, the low SES area covered 80% of the town, and the high SES area covered 20%. There were 900 households in the low SES area and 100 households in the high SES area. Therefore, the high SES area is less dense (0.5households/1000 square units[20]), than the low SES area (1.125

---

[19] Probability of selection for a sampled household in the standard town = $\frac{\pi r^2}{A} * \frac{1}{HH_S}$, and the probability of selection for a sampled household in the dense town = $\frac{4\pi r^2}{A} * \frac{1}{4HH_S} = \frac{\pi r^2}{A} * \frac{1}{HH_S}$

[20] $\frac{100\ households}{(200*1000)\ units\ square\ area} * 1000$

households/1000 square units[21]). The sample consisted of 100 households, picked from a set of datapoints in which some circles had no households to sample.

If the probability of choosing an empty circle was the same in the high SES area as in the low SES area, then 20 households are expected to be sampled from the high SES area (20% * 100) and 80 from the low SES area (80% * 100), yielding a sample total of 100 households. This meant that 20% of the households in the high SES area ($\frac{20}{100} = 20\%$) would be chosen, but less than 10% of the households in the low SES area ($\frac{80}{900} = 8.9\%$) would be chosen. However, since the high SES area was less dense than the low SES area, there is a higher probability for a random datapoint selected in the high SES area not to include a household in its circle. This meant that there was a need to randomly pick new datapoints until there was one with at least one household that could be sampled. Given the above, that "successful" datapoint has a higher probability of being chosen from the low SES area. Therefore, the percentage of households in the low SES area that are chosen is expected to be higher than 8.9%, and the percentage of households in the high SES area that are chosen is expected to be lower than 20%. So the probability of choosing a household from the high SES area versus the low SES area is affected by town density (i.e. the effect of empty circles in the low dense areas). And for any given sample, the number of households chosen from the high and low SES areas is affected by a combination of town density and the sampling variability.

E. Standard Error Obtained Via Theory and Simulations

Standard error can be computed in two different ways. One is using what theory tells us about standard errors, and the other is through simulations.

1. Through theory:

As per theory and the CLT, if a sufficiently large number of samples of size n are drawn from a population with a normal or non-normal distribution, where n is large enough, and $\bar{x}$ = sample mean and SE($\bar{x}$) = standard error of sample mean, then:

- The sampling distribution of the sample means will be asymptotically a normal distribution as will the sampling distribution of the means of sample means.

[21] $\frac{900\ households}{(800*1000)\ units\ square\ area} * 1000$

- E(mean of means) = E(mean of population), i.e. $E(\bar{\bar{x}}) = \mu$.
- SE(x) is calculated directly from n_s sample means taken from the distribution of sample means i.e. standard deviation of sample means
- E(SE) = Square Root of Average of n_s SE^2, i.e. $E(SE) = \sqrt{\overline{SE^2}}$).

2. Through simulations:

In the simulations, n_s samples were obtained from the population, of size n each. For each sample, the weighted mean and standard error for income and gun ownership were calculated, as per the formulas outlined in the Sampling Theory section. The following three calculations were then done for each set of n_s iterations:

- Calculated the mean of sample means
- Calculated the standard deviation of sample means
- Calculated the mean of variances, i.e. $\overline{SE^2}$

Using what has been discussed above, to check if the SE obtained from the simulations equates to the SE defined as per theory, the following is required using variances:

$E(SE^2)$ = Variance (Sample Means)

→ $E(\frac{s^2}{n})$ = Variance $(\bar{x})$

→ Mean(SE^2's) = Variance (Sample Means)

→ Average $(\frac{s^2}{n})$ = Variance $(\bar{x})$

→ $\overline{SE^2}$ = Variance $(\bar{x})$

In other words, if the mean of variances (i.e. $\overline{SE^2}$) equals the variance of sample means, this implies that the issue of different probabilities existing for the selection of a household does not affect the point estimates. However, when looking at the results from the simulations, I will be using SE (i.e. $\sqrt{\overline{SE^2}}$) and SD $(\bar{x})$ since it is easier to work with smaller numbers, hence expect $\sqrt{\overline{SE^2}}$ = SD $(\bar{x})$.

5. SIMULATION SETUP & RESULTS

In this section, I will provide the town and sample parameters, town plans and graphs of income distribution for both towns. I will also provide the tables, charts and graphs which show the results from the simulations, and discuss those before drawing a conclusion from all the available information.

A. Simulation Inputs and Town Plan for the Standard Town

In my simulations, I made certain decisions about how to create the "standard" and dense town. I tried to make some of the parameters reasonably comparable to what was observed in the South Lebanon study. I had a low and a high SES area in the town, and households a reasonable distance apart in both areas. I tried to ensure a reasonable distribution of parameters in both SES areas, including the representation of the continuous variable, income. However, I did have to make some simplifications in my simulations but in practice, certain aspects of it were reasonably realistic while others were not. In such a complex situations, the simulation had to be at least somewhat 'theoretical'.

When creating each of the two towns, I provided inputs. The inputs have been defined below for the low and high SES areas separately in Tables 5.1a and 5.1b, respectively:

	Low SES Area Inputs		
Number of Buildings	900		
Minimum Distance between Buildings	5		
Distance from Boundary	0		
Low SES Area Size (L x W)	800	x	1000
Distance from West Boundary	201		
Percentage with Characteristic (e.g Gun Ownership)	10.00%		
Mean of Income (units)	6		
Variance of log Income (units)	1.0		

Table 5.1a: Inputs for the low SES area.

	High SES Area Inputs		
Number of Buildings	100		
Minimum Distance between Buildings	10		
Distance from Boundary	0		
High SES Area Size (L x W)	200	x	1000
Distance from West Boundary	0		
Percentage with Characteristic (e.g Gun Ownership)	7.00%		
Mean of Income (units)	9		
Variance of log Income (units)	2.3		

Table 5.1b: Inputs for the high SES area.

Once the standard town had been created using the defined inputs, its population parameters were as below (in Tables 5.2a and 5.2b), including mean, standard deviation and standard error (SE)[22]:

Income

Income Mean	2,321
Income Std. Dev	11,698
Income SE of Mean	1,109

Table 5.2a: Population parameters, including mean, standard deviation and standard error (SE), for income.

Gun Ownership

Gun Ownership Proportion (p)	0.0970
Gun Ownership Std. Dev	0.2961
Gun Ownership SE of Mean	0.0281

Table 5.2b: Population parameters, including mean, standard deviation and standard error (SE), for income.

[22] Note the standard errors are for a sample of 100 observations, and are expected values with the FPC applied.

As per Figure 5.1, this is what the town plan looked like for the standard town:

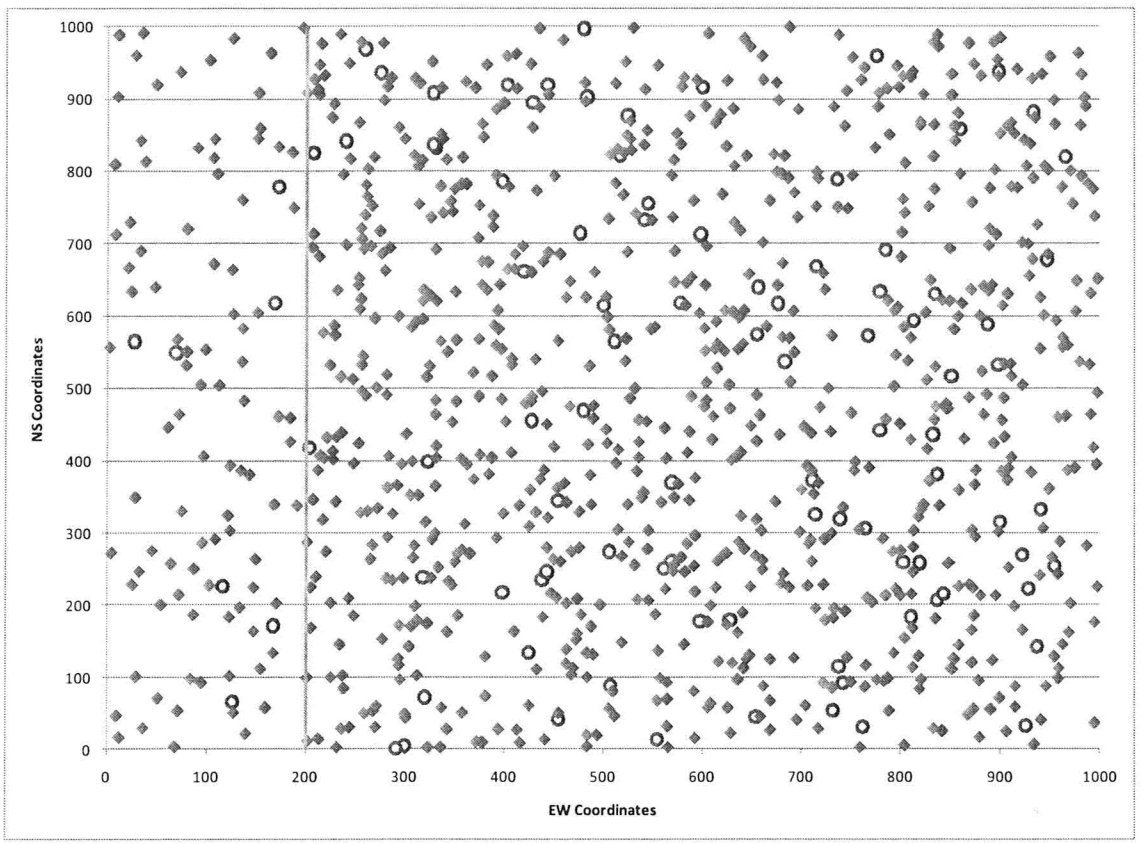


Figure 5.1: This is the town plan for the standard town, with the 97 red circles being the households of gun owners and the rest blue diamonds being all other households without a gun.

By looking at the town plan, it is easy to see that the town density is different in both areas, with the high SES area being less dense than the low SES area. In this plan, 1000 households are placed in a town area of 1,000,0000 square units (1000 x 1000 units).

B. Simulation Inputs and Town Plan for the Dense Town

The inputs for the dense town were the same as for the standard town with the sole exception that the length and width of the town were halved, so the town was one quarter of the size of the standard town. The relative sizes of the high SES area (area H) and the low SES area (area L) were the same as in the standard town (1:4 ratio). The inputs for the dense town are in Tables 5.3a and 5.3b below:

	Low SES Area Inputs		
Number of Buildings	900		
Minimum Distance between Buildings	5		
Distance from Boundary	0		
Low SES Area Size (L x W)	400	X	500
Distance from West Boundary	101		
Percentage with Characteristic (e.g Gun Ownership)	10.00%		
Mean of Income (units)	6		
Variance of log Income (units)	1.0		

Table 5.3a: Inputs for the low SES area.

	High SES Area Inputs		
Number of Buildings	100		
Minimum Distance between Buildings	10		
Distance from Boundary	0		
High SES Area Size (L x W)	100	X	500
Distance from West Boundary	0		
Percentage with Characteristic (e.g Gun Ownership)	7.00%		
Mean of Income (units)	9		
Variance of log Income (units)	2.3		

Table 5.3b: Inputs for the high SES area.

Once the dense town had been created using the defined inputs, its population parameters were as below (in Tables 5.4a and 5.4b), including mean, standard deviation and standard error (SE)[23]:

Income

Income Mean	2,836
Income Std. Dev	13,016
Income SE of Mean	1,234

Table 5.4a: Population parameters, including mean, standard deviation and standard error (SE), for income.

23 Note the standard errors are for a sample of 100 observations, and are expected values with the FPC applied.

Gun Ownership

Gun Ownership Proportion (p)	0.0970
Gun Ownership Std. Dev	0.2961
Gun Ownership SE of Mean	0.0281

Table 5.4b: Population parameters, including mean, standard deviation and standard error (SE), for income.

As per Figure 5.2, this is what the town plan looked like for the dense town:

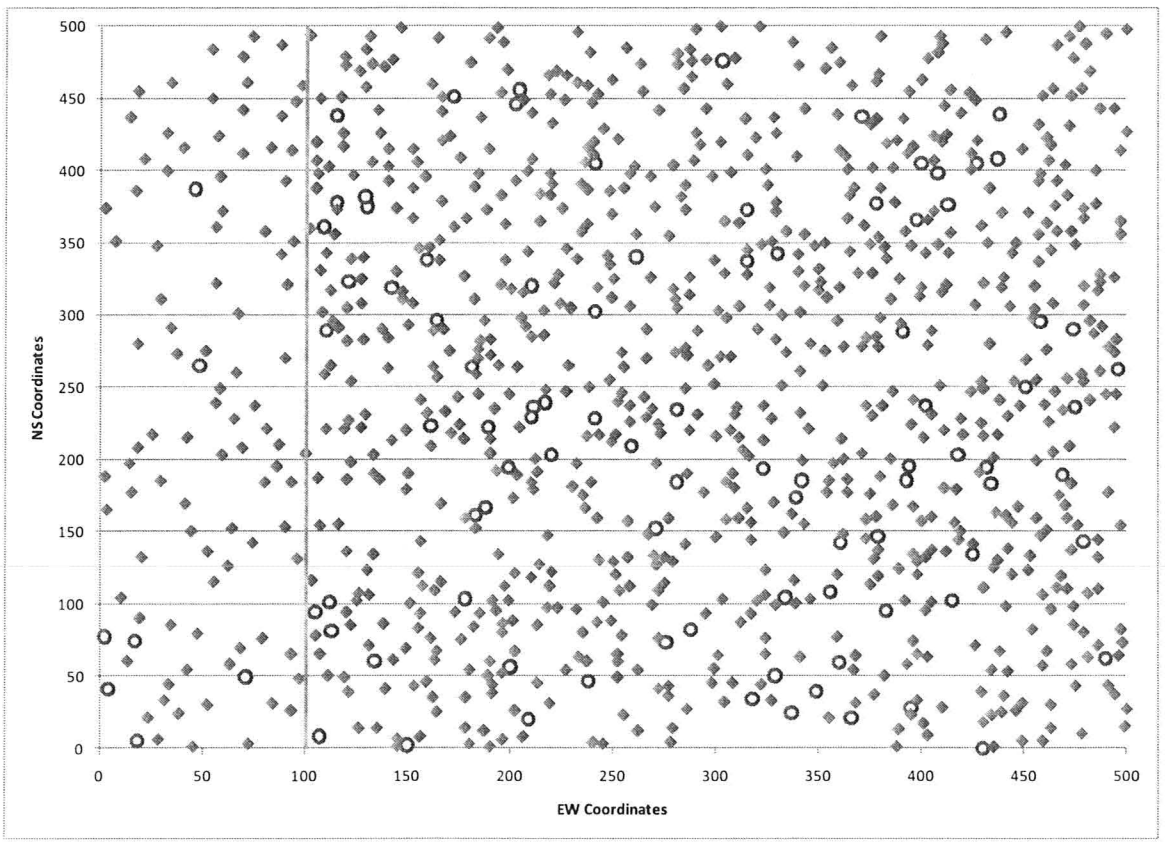


Figure 5.2: This is the town plan for the dense town, with the 97 red circles being the households of gun owners and the rest blue diamonds being all other households without a gun.

Similar to what was observed in the standard town plan, the town density is different in both areas in the dense town plan. Again, the high SES area is less dense than the low SES area since the households are packed more closely in the

low SES area, assuming majority of the residents are low income earners. In this plan, 1000 households are placed in a town area of 250,000 square units (500 x 500 units). Compared to the standard town, the dense town is ¼ the size of it.

C. Sample Inputs for the Standard and Dense Town

Once the town was created, many samples of size n were drawn from it. The household sampled from each selected datapoint had to lie within a circle around the datapoint of a specified radius. I made certain decision about what inputs I would use when drawing my samples. The sample inputs for any sample drawn from the standard or the dense town, respectively, are as outlined in Tables 5.5a and 5.5b below:

Sample Inputs			
Sample Size	100		
Datapoint Radius	20		
Distance from Boundary	0		
Town Size (L x W)	1000	x	1000
Distance from West Boundary	200		

Table 5.5a: Inputs for selecting any sample from the standard town.

Sample Inputs			
Sample Size	100		
Datapoint Radius	20		
Distance from Boundary	0		
Town Size (L x W)	500	x	500
Distance from West Boundary	100		

Table 5.5b: Inputs for selecting any sample from the dense town.

D. Income Histogram for the Standard and Dense Town

A log-normal distribution was proposed for household income in each area (H and L) of both the standard and dense towns. Therefore, the income histogram for all households in the standard and dense town (shown in Figures 5.3 and 5.4, respectively), shows that the distribution of income is a mixture of two log-normal distributions, and looks something like this:

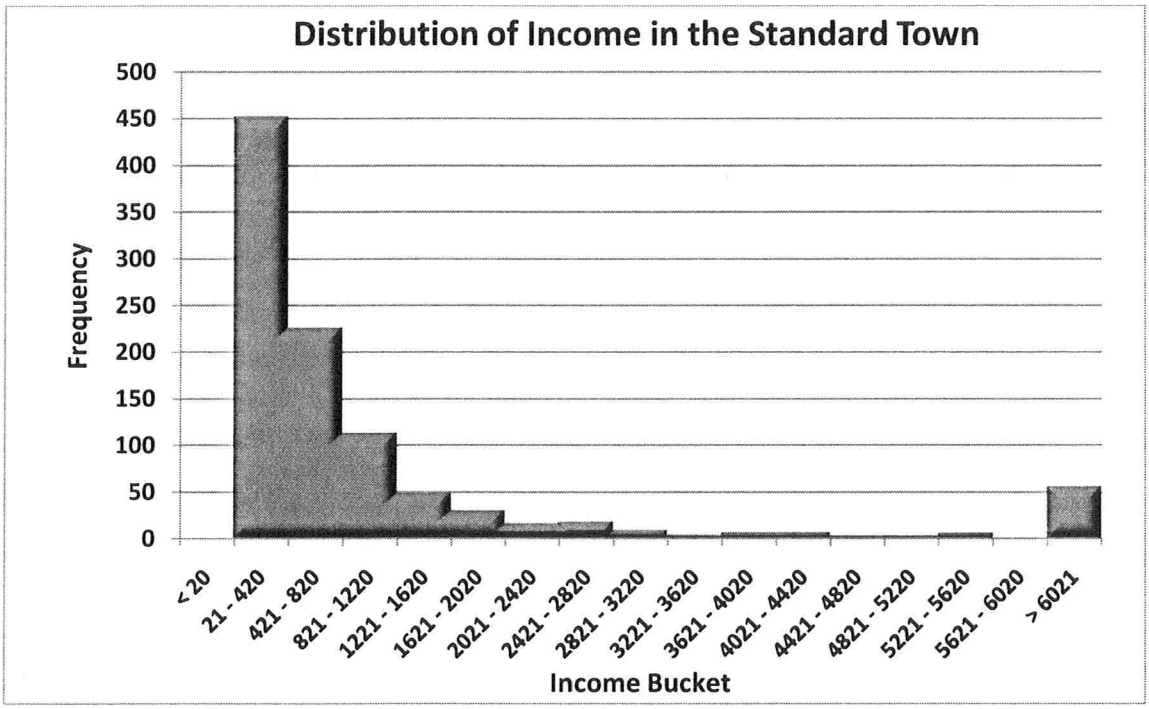


Figure 5.3: Income histogram for the standard town.

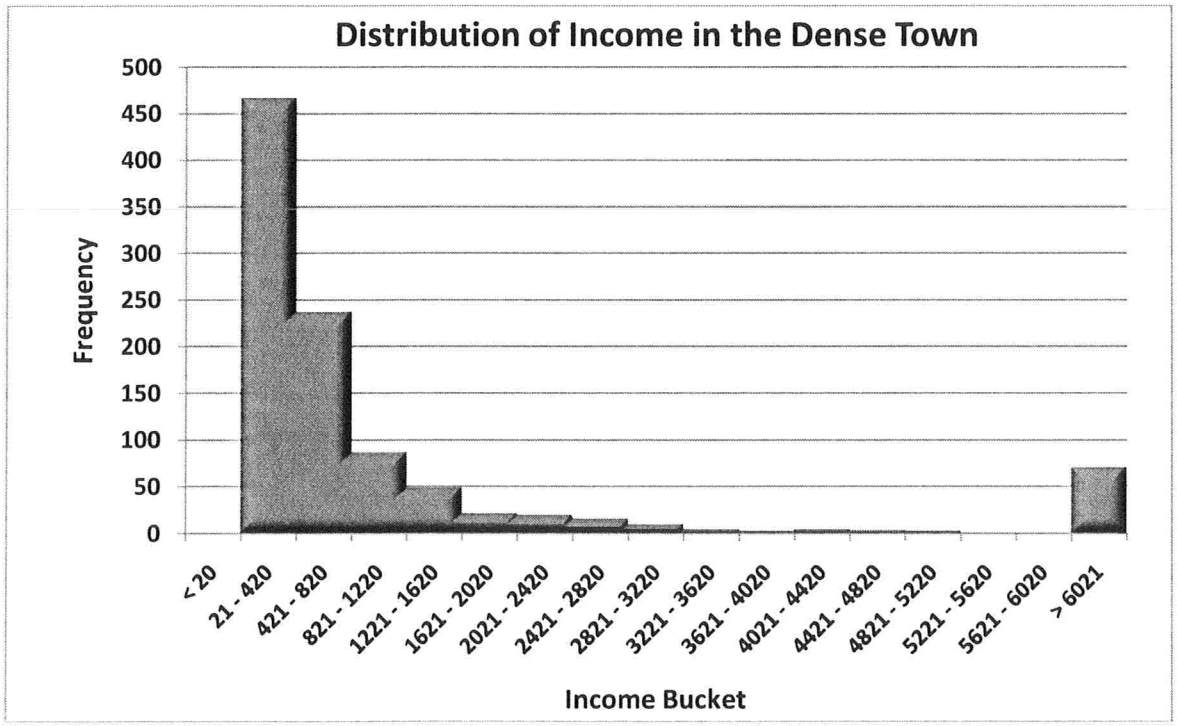


Figure 5.4: Income histogram for the dense town.

When setting up the population, I randomly sampled 1000 values from two normal distributions. 900 values were from a normal distribution assumed for the low SES area and the other 100 values were from a normal distribution assumed for the high SES area. This was to generate the distribution of income for the high and low income SES in the town (the actual income values were the exponent of the values sampled from the normal distribution). The income histograms have a roughly log-normal shape (as expected) in both towns. However, there is a slight bump between ranges of 2421-2820 and 3621-4420 in the standard town histogram, and between the range of 4021-4820 in the dense town histogram, presumably due to random variability when sampling values from the normal distributions. Also, note that the graphs do spike up at points where income > 6021 in both towns, because that range encompasses most, if not all, of the households in the high SES area. If the income histograms for households in the low and high SES areas of both towns were sketched individually, then they would match a log-normal graph as shown in Figures 5.5, 5.6, 5.7 and 5.8 below:

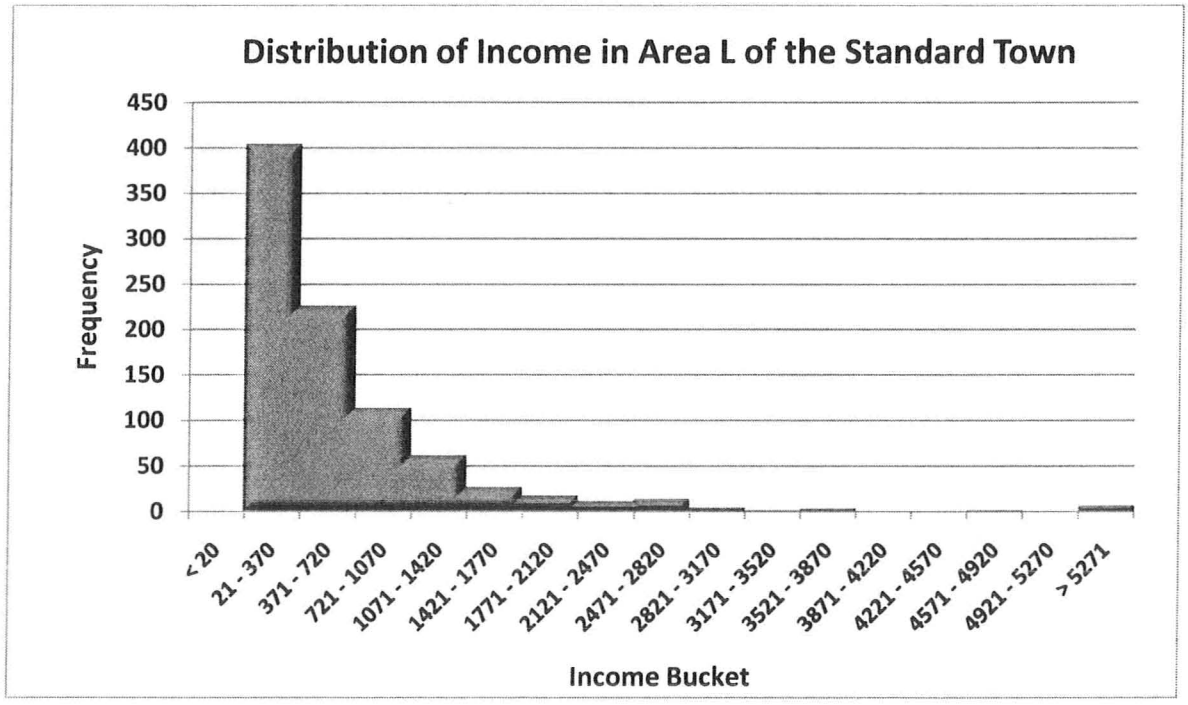


Figure 5.5: Income histogram for area L in the standard town.

In the standard town, income in Area L is roughly log normally distributed, with very few households falling in higher income brackets e.g. income > 5271.

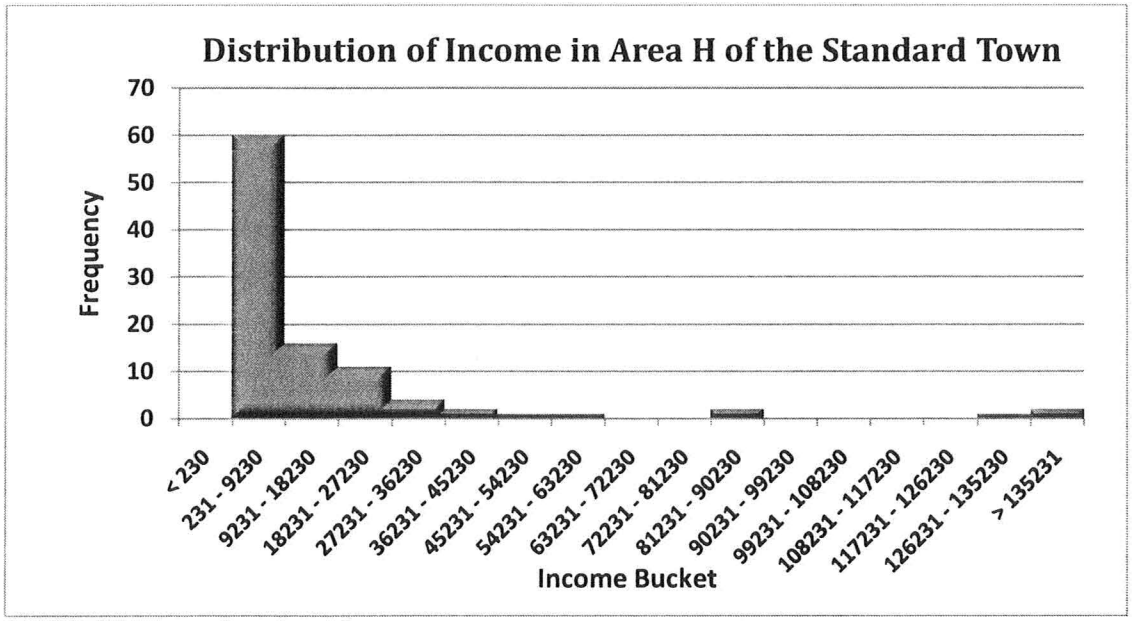


Figure 5.6: Income histogram for area H in the standard town.

In the standard town, the income in Area H is also roughly log normally distributed with some bumps along the way, e.g. between ranges of 81231-90230 and > 126231, due to the very large range within which income is distributed in the high SES area households.

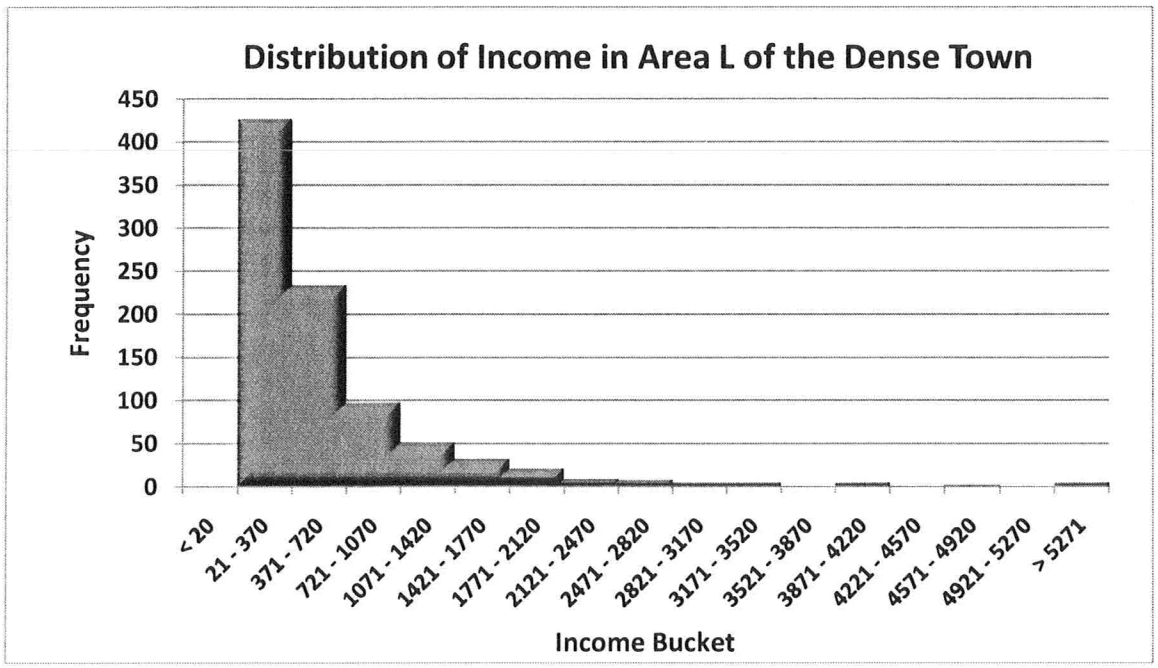


Figure 5.7: Income histogram for area L in the dense town.

In the dense town, income in area L is roughly log normally distributed, again with very few households falling in the higher income brackets e.g. in the brackets of 3871 - 4920 and > 5271.

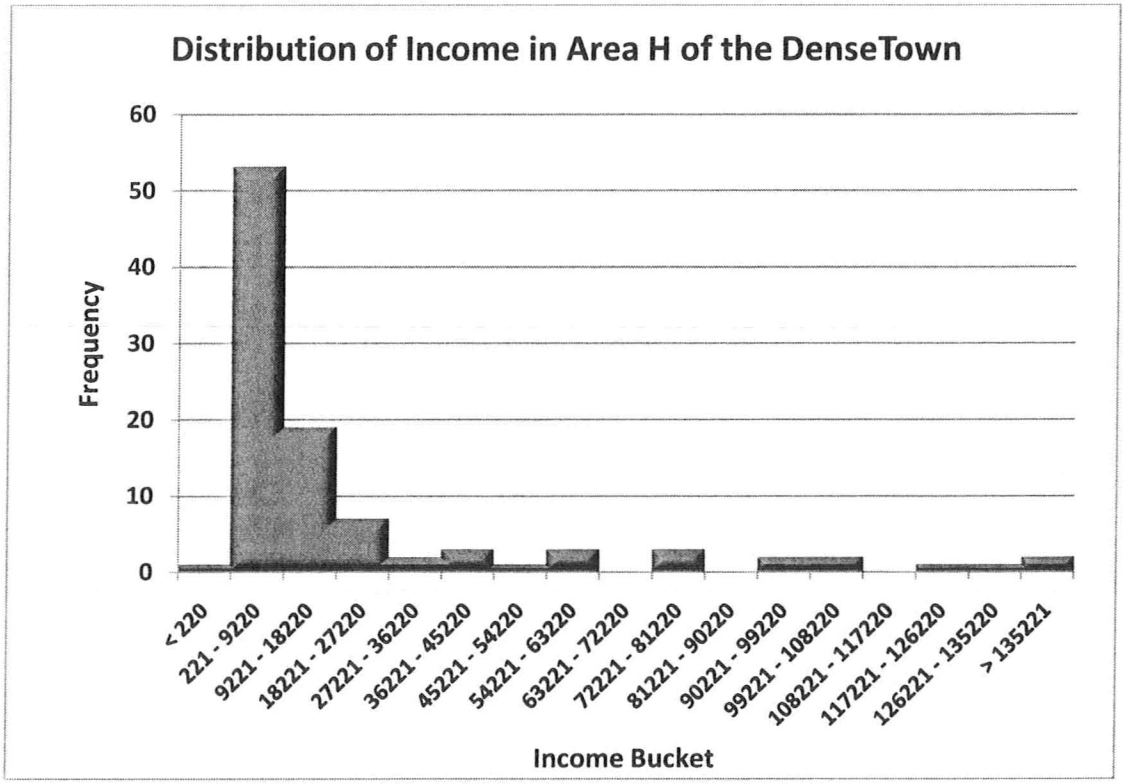


Figure 5.8: Income histogram for area H in the dense town.

In the dense town, the income in area H is also roughly log normally distributed with many more some bumps along the way, e.g. between ranges of 54221-63220, 72221- 81220, 90221-108220 and for income > 135221, again due to the very large range within which income is distributed in the high SES area households.

E. Tables and Plots for the Standard and Dense Town

Once each town was created, five sets of n_s samples were drawn from the population of each town, where n_s = 2000, 4000, 6000, 8000 and 10000 samples (or iterations), respectively. Each sample contained 100 households. For each set of n_s samples, the three estimates shown in Table 5.6 were calculated for income and gun ownership.

Number	Estimate Calculated
1	Mean of n_s sample means
2	Standard deviation of n_s sample means
3	C = Estimated standard error = E(SE) = $\sqrt{}$(Mean of n_s standard errors^2), where the SE was calculated separately for each of the n_s samples.

Table 5.6: Estimates calculated for sampled data for the standard and dense towns

STANDARD TOWN:

The results for the estimates outlined in Table 5.6 are presented in Tables 5.7 and 5.8 below for the standard town:

Table 1: Income			
Iterations	**Mean of Means**	**SD of Means**	**C = E(SE)**
2,000	2,301	962	1,116
4,000	2,279	975	1,105
6,000	2,333	1,017	1,148
8,000	2,322	994	1,134
10,000	2,318	996	1,131
Expected Population Values	**2,321**	**1,109**[24]	

Table 5.7: Estimates of the mean of means, standard deviation of means and C, for iterations of 2000, 4000, 6000, 8000 and 10000, for income.

Table 2: Gun Ownership			
Iterations	**Mean of Proportions**	**SD of Means**	**C = E(SE)**
2,000	0.0973	0.0319	0.0282
4,000	0.0976	0.0323	0.0283
6,000	0.0966	0.0320	0.0281
8,000	0.0966	0.0320	0.0281
10,000	0.0968	0.0313	0.0282
Expected Population Values	**0.0970**	**0.0281**[25]	

Table 5.8: Estimates of the mean of means, standard deviation of means and C, for iterations of 2000, 4000, 6000, 8000 and 10000, for gun ownership.

[24] Theoretical SE of mean = $\sigma/\sqrt{n}$ * FPC

[25] Theoretical SE of mean = $\sigma/\sqrt{n}$ * FPC

Plots were also created for both towns, which used the information contained in the tables above[26]. Estimates of the mean of n_s sample means, standard deviation of n_s sample means, and C were graphed against the number of iterations[27], for income and gun ownership. The plots are shown below.

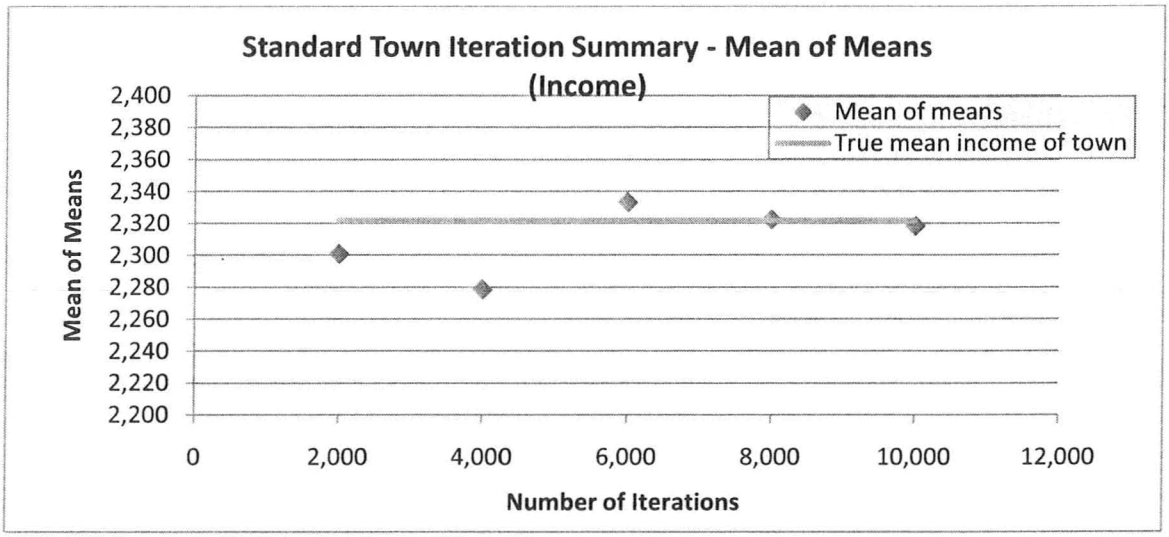


Figure 5.9: Plot of the mean of means against the number of iterations (2000, 4000, 6000, 8000 and 10000), for income.

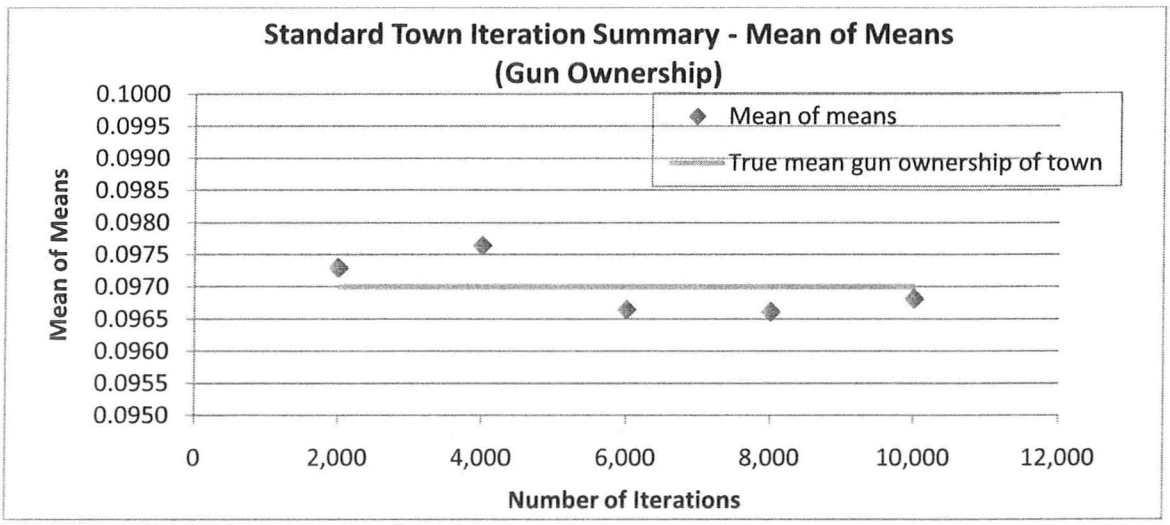


Figure 5.10: Plot of the mean of means against the number of iterations (2000, 4000, 6000, 8000 and 10000), for gun ownership.

[26] Note that the scales have been kept large for clarity purposes when presenting all the plots for the standard and dense towns.

[27] Number of iterations = 2000, 4000, 6000, 8000 and 10000

The results from the simulations found that the mean of means for both income and gun ownership are close to the μ values of 2,321 and 0.097 respectively, with the mean of means for all sets of iterations falling just above and below the true mean.

Given that C is defined as $\sqrt{(\text{Mean of } n_s \text{ standard errors}^2)}$, plots of the standard deviation of means and C against the number of iterations are shown below in Figures 5.11 and 5.12, for the standard town.

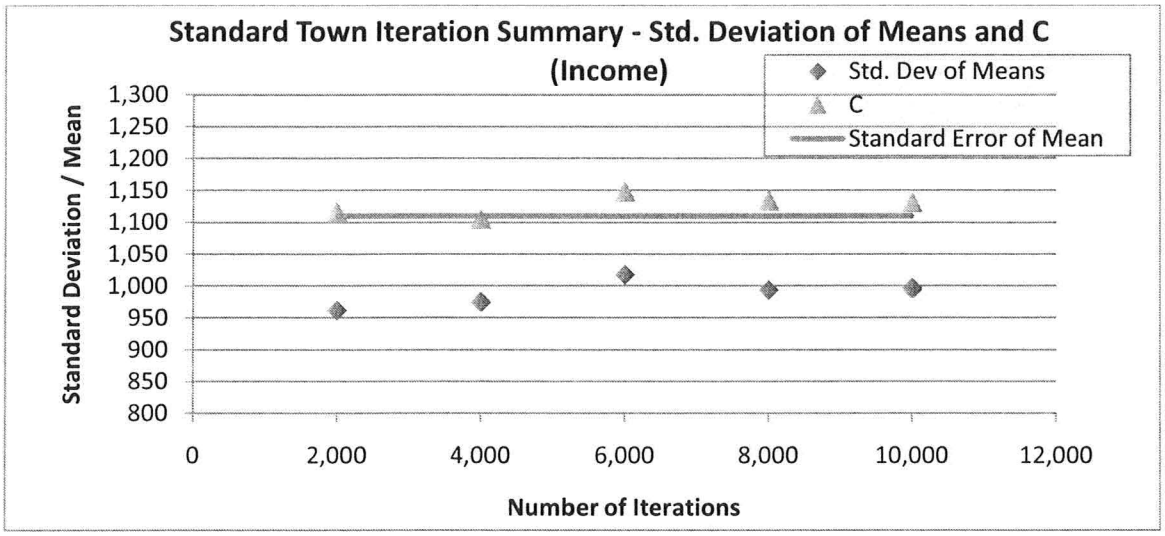


Figure 5.11: Plot of the standard deviation of means and C against the number of iterations (2000, 4000, 6000, 8000 and 10000), for income.

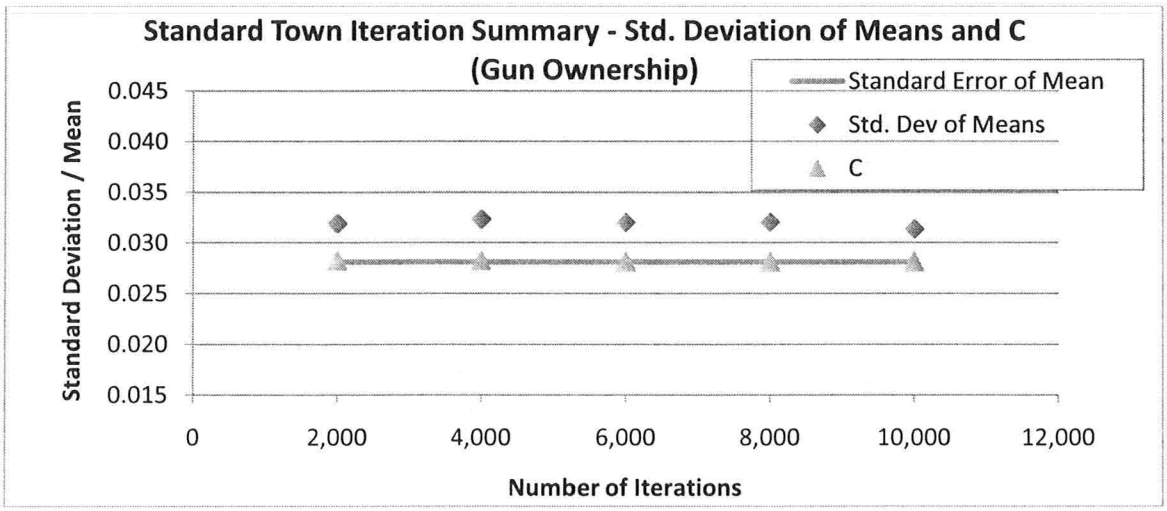


Figure 5.12: Plot of the standard deviation of means and C against the number of iterations (2000, 4000, 6000, 8000 and 10000), gun ownership.

C is very close to the theoretical standard error of the mean (which is based on the known standard deviation of income for the town population) for both income and gun ownership of 1,110 and 0.0281 respectively. The standard deviation of means is also close to C for both income and gun ownership, with a greater difference observed between these two estimates versus the difference observed between C and standard error of the mean. However, the standard deviation of means consistently falls below C for income, and falls above C for gun ownership. This seems to suggest something more systematic going on, which could be due to the sampling method used.

DENSE TOWN:

The results for the three estimates outlined in Table 5.6 (mean of n_s sample means, standard deviation of n_s sample means, and C) are presented in Tables 5.9 and 5.10 for the dense town below:

Table 1: Income			
Iterations	**Mean of Means**	**SD of Means**	**C = E(SE)**
2,000	2,755	946	1,212
4,000	2,769	964	1,223
6,000	2,773	983	1,226
8,000	2,757	959	1,219
10,000	2,757	965	1,218
Expected Population Values	**2,836**	**1,234**[28]	

Table 5.9: Estimates of the mean of means, standard deviation of means and C, for iterations of 2000, 4000, 6000, 8000 and 10000, for income.

Table 2: Gun Ownership			
Iterations	**Mean of Proportions**	**SD of Means**	**C = E(SE)**
2,000	0.0966	0.0311	0.0281
4,000	0.0972	0.0307	0.0282
6,000	0.0968	0.0312	0.0282
8,000	0.0970	0.0313	0.0282
10,000	0.0971	0.0315	0.0282
Expected Population Values	**0.0970**	**0.0281**	

Table 5.10: Estimates of the mean of means, standard deviation of means and C, for iterations of 2000, 4000, 6000, 8000 and 10000, for gun ownership.

[28] Theoretical SE of mean = $\sigma/\sqrt{n} * FPC$

Similar to what was done in the standard town, plots were created for the dense town, which used the information contained in Tables 5.9 and 5.10 above. Estimates of the mean of n_s sample means, standard deviation of n_s sample means, and C were graphed against the number of iterations[29], for income and gun ownership. The plots are shown below.

Plots of the mean of means against the number of iterations are shown in Figures 5.13 and 5.14:

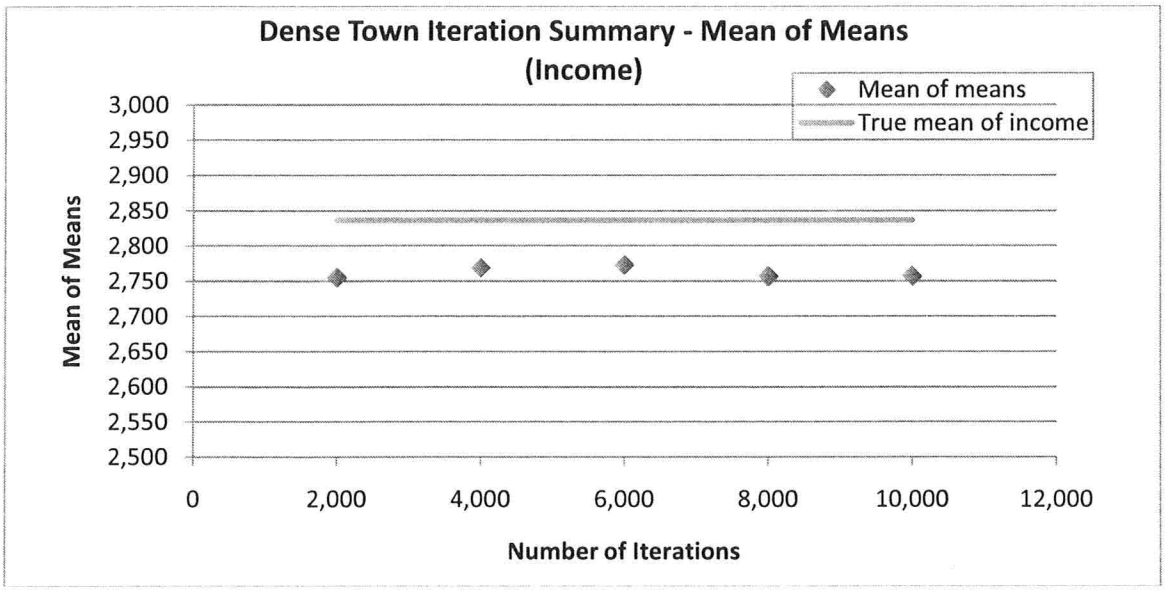


Figure 5.13: Plot of the mean of means against the number of iterations (2000, 4000, 6000, 8000 and 10000), for income.

[29] Number of iterations = 2000, 4000, 6000, 8000 and 10000

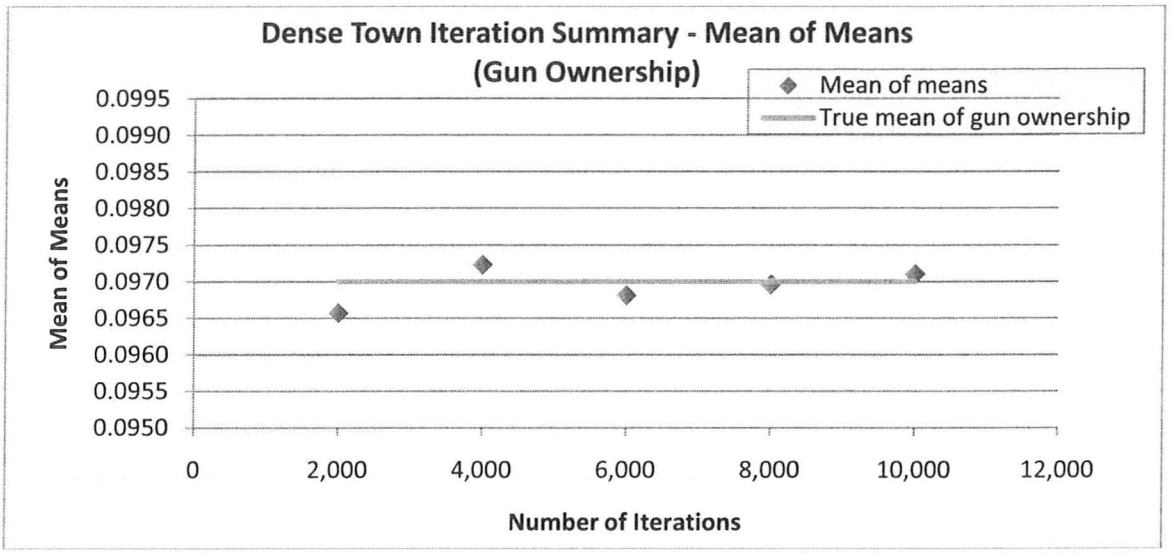


Figure 5.14: Plot of the mean of means against the number of iterations (2000, 4000, 6000, 8000 and 10000), for gun ownership.

The results from the simulations found that the mean of means for both income and gun ownership are close to the true μ of 2,836 and 0.097 respectively. Similar to what was observed in the standard town plots, the mean of means for all sets of iterations fall just above and below the true mean for gun ownership. However, the mean of means are below the true mean for all sets of iterations for income (and close in value to each other). This seems to suggest something more systematic going on, which could be due to the sampling method used. It should be noted that there are two issues that arise with this method, including different probabilities of selection and overlapping circles. Although they are two distinct issues, since the code in the simulations does not really account for either of these issues, a combination of them can cause the results above, especially since in the dense town there is likely to be more of an issue with running into overlapping circles.

Given that C is defined as $\sqrt{}$(Mean of n_s standard errors^2), plots of the standard deviation of means and C against the number of iterations are shown below in Figures 5.11 and 5.12, for the dense town.

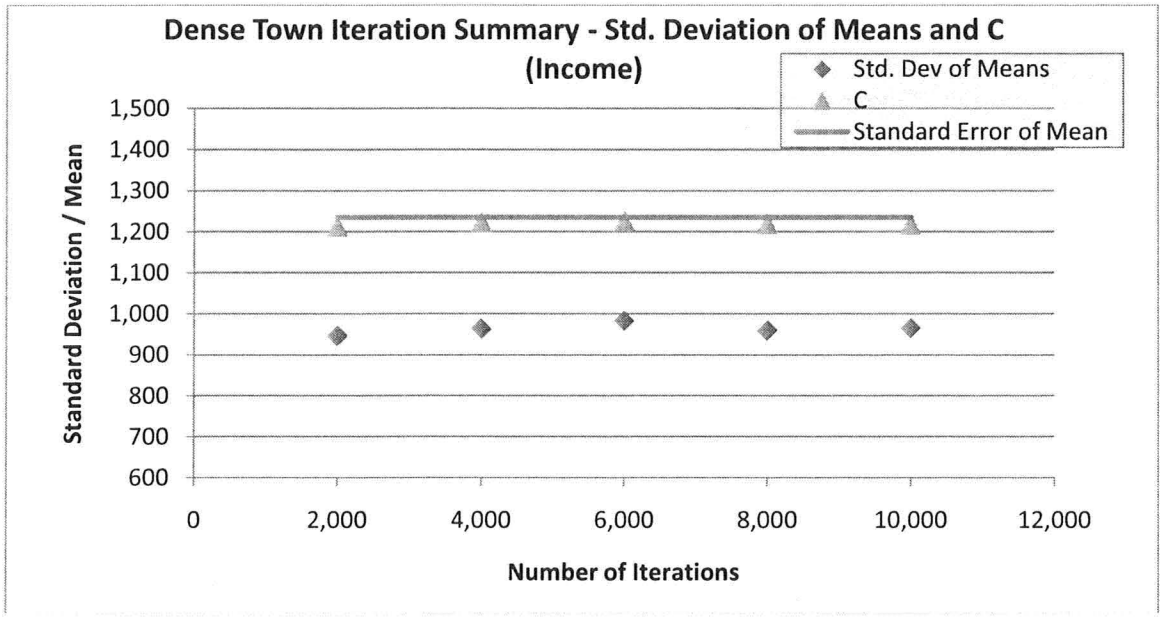


Figure 5.15: Plot of the standard deviation of means and C against the number of iterations (2000, 4000, 6000, 8000 and 10000), for income.

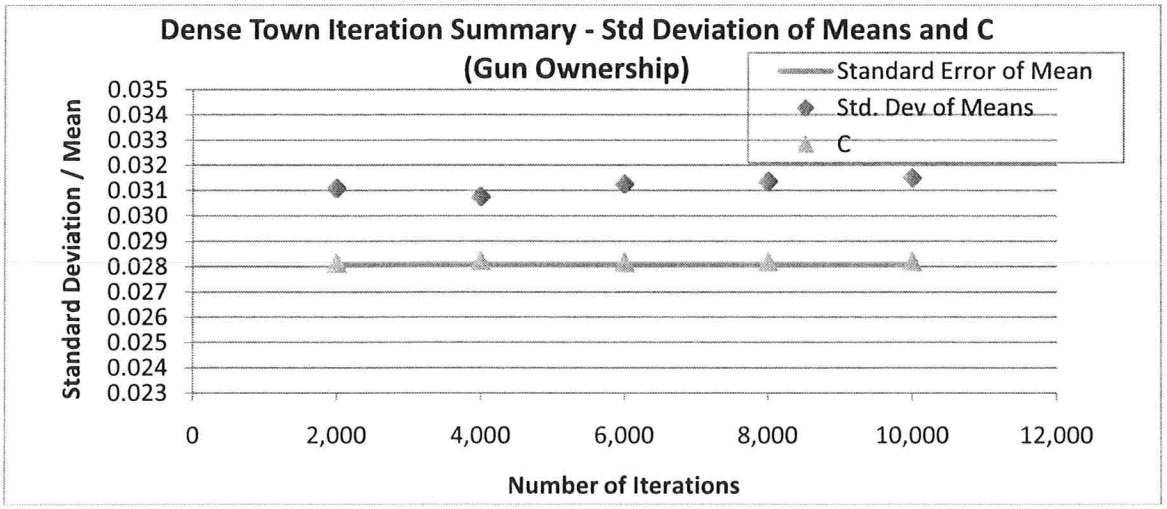


Figure 5.16: Plot of the standard deviation of means and C against the number of iterations (2000, 4000, 6000, 8000 and 10000), for gun ownership.

The results obtained in the dense town are very similar to what was observed in the standard town. The values of C are very close to the theoretical standard errors of the mean for both income and gun ownership of 1,235 and 0.0281 respectively. The standard deviation of means is also close to C for both income and gun ownership, with a greater difference observed between these two

estimates versus the difference observed between C and standard error of the mean. Moreover, as was observed with the standard town data, the standard deviation of means consistently falls below C for income, and falls above C for gun ownership with the dense data too. Given this systematic occurrence in both towns, one may assume that this could be due to the sampling method used. As discussed earlier, it may be due to the fact that the code used in the simulations does not really account for the two issues inherent in this sampling method (different probabilities of selection and overlapping circles), and a combination of them can cause the results seen in the plots above.

Another analysis was also done on the n_s sample means obtained in both the standard and dense town simulations (where n_s = 2000, 4000, 6000, 8000 and 10000 iterations or samples drawn from the population). The aim was to count how many of the n_s sample means were within 1.96 standard errors of the true mean, for income and gun ownership. This is because 95% of the area under the curve of a normal distribution lies within approximately 1.96 standard errors of the mean. This assumption is based on the fact that the central limit theorem proves that the distribution of sample means for samples of size 100 is very close to a normal distribution. In expectation, 95% of the sample means should be within 1.96 standard errors of the true mean. The results varied somewhat (but not a lot) depending on the number of iterations run.

For the standard town, the results for the 2000, 4000, 6000, 8000 and 10000 iterations, for income and gun ownership are shown in table 5.11:

	Proportion of Sample Means within 1.96SE's of True Mean			
	Income		**Gun ownership**	
# Iterations	**Count**	**Proportion**	**Count**	**Proportion**
2,000	1936	96.8%	1834	91.7%
4,000	3862	96.6%	3674	91.9%
6,000	5729	95.5%	5512	91.9%
8,000	7695	96.2%	7348	91.9%
10,000	9619	96.2%	9258	92.6%

Table 5.11: The proportion of sample means within 1.96 standard errors of the true mean for income and gun ownership, for iterations of size 2000, 4000, 6000, 8000 and 10000.

For income, the proportion of the n_s sample means that are within 1.96 SE's of the true mean is close to 96%, which is a little above the expected 95%. However, the proportion of the n_s sample means that are within 1.96 SE's of the true mean for gun ownership is lower than the expected 95% at around 92%. In other words, there appears to be less variance than that expected for the income observations, given that roughly 96% of the data lies within 1.96 SEs of the mean; and there appears to be more variance than expected for the gun ownership observations, given that only about 92% of the data lies within 1.96 SEs of the mean.

For the dense town, the results for the 2000, 4000, 6000, 8000 and 10000 iterations for income and gun ownership are shown in Table 5.12:

	Proportion of Sample Means within 1.96SE's of True Mean			
	Income		**Gun ownership**	
# Iterations	**Count**	**Proportion**	**Count**	**Proportion**
2,000	1979	99.0%	1847	92.4%
4,000	3938	98.5%	3724	93.1%
6,000	5890	98.2%	5548	92.5%
8,000	7880	98.5%	7394	92.4%
10,000	9839	98.4%	9252	92.5%

Table 5.12: The proportion of sample means within 1.96 standard errors of the true mean for income and gun ownership, for iterations of size 2000, 4000, 6000, 8000 and 10000.

In the dense town simulations, results again differed from expectations. For income, the proportion of the n_s sample means that are within 1.96SEs of the true mean was a little over 98%, rather than the expected 95%. The proportion of the n_s sample means that are within 1.96SEs of the true mean for gun ownership is lower than the expected 95% at around 92%. In other words, there appears to be much less variance than that expected for the income observations, given that roughly 98% of the data lies within 1.96SEs of the mean. And there appears to be more variance than that expected for the gun ownership observations, given that roughly only 92% of the data lies within 1.96SEs of the mean.

By looking at the income and gun ownership results of both the standard and dense town simulations, a few things have been observed:

First, the mean of means for income and gun ownership appear to be accurate as they are very close to the true mean, for each of the five sets of iterations, as was expected. In fact, as the number of iterations (n_s) increases, the mean of means appear to be closer to the true population mean (μ), except in the case of the mean of means for income in the dense town. This suggests that the sampling method is able to produce estimates of the population mean close to the true mean, with some random variability. With regards to the mean of means for income in the dense town, there may be something more systematic going on, given all estimates of the mean of means are slightly below the true mean. We cannot state with certainty if bias is present, since the true mean is still within 95% of the confidence interval. However, it could be an issue with the sampling method and the fact that it not always able to pick samples randomly enough, therefore resulting in biased results since all five means of means appear to be biased downwards. Or it could be due to overlapping circles, which is an issue found with this sampling method but a rather complex one to solve, so it has not been investigated in detail in the thesis.

Secondly, the standard deviation of means and the estimated standard error of mean (C) are both very close to the theoretical standard error of mean ($\sigma/\sqrt{n}$) for income and gun ownership. C is very close to the theoretical standard error, whereas the standard deviation of means is only slightly less close. This may be explained by the fact that sampling error exists, so one should allow for some margin of error. However, the results seem to suggest that C might be a better way for estimating the standard error, versus using the standard deviation of means.

Another set of observations found that the proportion of sample means within 1.96SE's of the mean (for income and gun ownership in both towns) were different than the expected 95%. For both towns, the proportions were higher than expected for income, and lower than expected for gun ownership sample means. However, the proportions for income in the standard town are still closer to 95% versus the proportions in the dense town at 98%. Whereas, the proportions for gun ownership in the standard and dense towns are further away from 95% at 90% and 92%, respectively. Accordingly, this seems to suggest that the sampling method may not be as accurate as it seemed when analyzing the results of the SD of the sample means and C, or when comparing the results of the mean of means and the populations mean above.

The issues of different probabilities or even overlapping circles, might account for these inconsistent results:

1. The primary issue that exists with this sampling method of there being multiple probabilities of selection for a household may also explain the differences observed between the results and the true population values. This issue is inherent in the sampling methods and occurs whether one is sampling from a low density or a high density town. In order to determine whether the cause of the results differing from expected estimates is due to just the issue of multiple probabilities of selection, one may create a simulation in which the town is large enough that overlapping can be minimized if not eliminated. The results may then give a better idea as to whether the differences are due to either or both of the issues inherent in the overlapping method, or something else.

2. The issue of overlapping circles appears to be inherent in the sampling method especially in the dense town. It may be a significant reason for the differences observed between the simulated results and true values, given that the overlapping circles were ignored when samples were drawn from the population of both towns. This was due to the sheer complexity of the task and I chose to not take account of the fact that some circles may overlap and thus a household may be chosen from a number of circles with different probabilities of selection. More details are given in Appendix E. The issue of overlapping circles may also explain differences between estimates found in the dense town and standard town, since overlapping would occur more in smaller, denser towns.

However, to fully investigate these issues and see to what degree they affect the results obtained from the simulations, future work may be done on them as outlined in the next section.

6. STRENGTHS/WEAKNESSES/EXTENSIONS

In this section, I will identify strengths, limitations and possible extensions of this work. I will then discuss the validity of the new sampling method by considering how it deals with the issue of differing probabilities.

A. Strengths

The major benefit of the simulations was that they allowed us to determine how reliable the new sampling method was, by comparing results with true population parameters. Since there is no closed form equation to estimate the probabilities of selection with this method or its effect on the estimates of parameters, the simulations were useful in providing results that could be compared against the true population parameters.

Moreover, the code is set up to allow for most parameters to be altered upon any run of the simulation. This makes it easier to look at a number of different sets of simulations and make comparisons among them to see the effect of different variables. In the simulations I ran, I only changed the town size and kept everything else constant but in practice, all the variables can be altered.

Another benefit of using simulations is that one can make simplifications or put restrictions on the parameters, which allows one to examine the effects of varying specific parameters. In the simulations I made many simplifications (as outlined in the Introduction), one of which was to ignore any overlap of circles. More on this assumption is explained below.

Finally, another advantage to doing the simulations is that they provide results that may be compared to variations of this sampling method (and indeed different methods of sampling) in the future. In other words, if the current sampling method is modified and simulations are run using the new "modified" method, results from both methods can be compared against each other, to see which method provides more accurate estimates of the population parameters.

B. Limitations

There were two major sets of restrictions that I put on the simulations. The first were simplifications I made when setting up the simulated towns, and the other was an assumption I used when determining the probabilities of selection for households.

I made a number of simplifications in my simulations, which were conscious choices that allowed me to create both the simulated standard and dense towns. I believe the simplifications were reasonable as I accounted for many characteristics used when setting up a town. Moreover, the simplifications helped me in constructing the towns more easily and without complicating them too much, while ensuring that the simulated towns were just a simplified version of actual towns in the South Lebanon study, so the sampling method could be used as intended (i.e. for a population on which limited information is available).

The first simplification I made was that I considered only one town at a time, while the original study included a number of towns. If I simulated sampling from more than one town, many more variables would have to be dealt with e.g. town size, town density, characteristics of each town or village. Using one town made it much simpler as I was only working with a few variables. Another assumption I made was that each building contained only one household, whereas in reality some of the sampled buildings contained multiple residences, especially those in the low SES areas.

Another possible limitation was that I created the high and low SES areas of the town as rectangles next to each other, as shown in Figure 6.1 below. In this layout of the town, only datapoints close to the boundary have a possibility of sampling a low SES household through a datapoint in the high SES area.

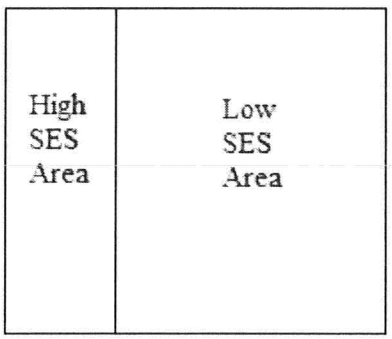


Figure 6.1: Layout of the town containing one low and one high SES area.

Now, I constructed the high and low SES areas of the town separately on one part of the grid, because it made it easier to sample households by defining both areas so specifically, since I knew exactly what boundaries I was sampling within. But in reality there could be several high and low SES areas in various parts of the town, as shown in Figure 6.2 below.

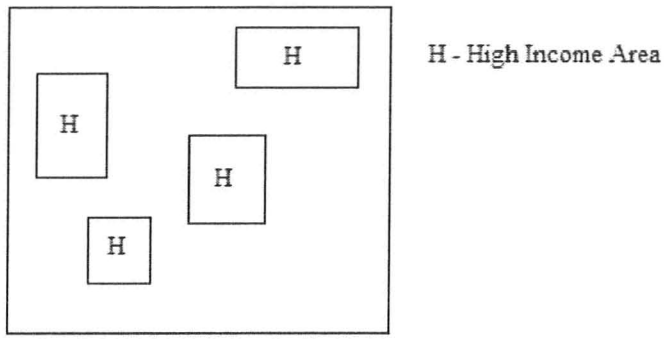


Figure 6.2: Town layout with one low and several high SES areas.

In this layout of the town, the total "length" of the boundary between the high and low SES areas is much longer compared to the boundary I have used in the simulated towns (Figure 6.1), since it is a sum of the boundaries of all four high SES areas. Accordingly, a much larger number of datapoints have a possibility of sampling a low SES household through a datapoint in the high SES area, given the boundary is much longer. However, it is very difficult to determine the exact effect of simulating a mixed town on the mean and standard deviation, and for that reason, I kept the town simple by assuming the high and low SES areas were contained in two blocks alongside each other.

Another simplification I used was that I kept all the variables consistent between both the simulations, except town size, to make it easier to analyze my results and determine the accuracy of the sampling method. Any differences between both sets of results were then just due to different town densities (and perhaps random variability in the creation of the towns).

When determining the probabilities of selection for households, I ignored the fact that some datapoint circles overlapped with each other in the sample, although in practice one would also observe the occurrence of overlapping circles and allow them, given it is something that is inherent with the sampling method. In the simulations, I was aware that a significant amount of overlapping took place in the sample based on a test included in the code, which meant that there were multiple probabilities of selection for households falling in an area of overlap. However, being able to extract those probabilities for the affected circles was rather complex and something I could not account for in my simulation. Consequently, I assumed that I could treat each individual circle as if it did not overlap with any other circle. One could wonder whether ignoring the overlapping issue leads to oversimplification and whether that is a problem. However, that risk had to be taken since the difficulty in dealing with the issue was rather profound. Further details about this issue are in Appendix E, although it shows the different probabilities to account for when only two circles overlap. One can then imagine

how difficult the task would be in calculating probabilities if more than two circles overlap with each other. Future work on this issue may be done to see if it can be resolved so that the results derived from this sampling method match expectations.

C. Extensions

If further work were done, there are a number of ideas that could be pursued. For the most part, all the simplifications mentioned above could be explored so that the variables kept constant in my simulations could be varied. In fact, a greater number of simulations could be run with slight variations in each simulation (i.e. by changing variables) to create new and altered towns. The parameter estimates obtained from those simulations could then be compared against the true parameters to determine which variable had the greatest impact. My analysis was done on two simulations, a "standard town" and a denser town, and I only changed the town size between the two simulations. Other changes could include the number of households in the town, the relative sizes of high SES and low SES areas, or the income distributions. As well, other characteristics could be added in creating the towns or decreasing the samples. For simulations where multiple variables will be changed, note that there will be some variables that are related to or dependent on each other, e.g. town density depends on the minimum distance between buildings, or town density depends on the number of buildings in town etc. Aspects of the sample can also be changed, such as altering the sample size or increasing/ decreasing the datapoint radius. Moreover, a more complex extension would be to create a simulation which would consist of multiple towns with different variables in each town. Right now my simulation only had one town, but in order to somewhat replicate a real situation, a simulation with multiple towns would be needed.

Another extension would be to investigate the issue of overlapping circles. An effective solution might be one that could identify all the probabilities of selection for households falling in overlapping circles, and then compute a weighted average of those probabilities. However, the complexity of this task might be rather high, especially in cases where the simulated town might be physically very small but a large sample is drawn from it, thus creating a situation where many circles overlap each other. Another solution may be to create a town with tiny datapoint circles (so there is only 1HH / circle) to reduce the issue of multiple probabilities. The estimates obtained from the simulation could then be compared to the true population values and assessed whether any differences are due to the overlapping issue (and how significant they are) or something else.

In addition, there is another interesting problem that arises when overlapping circles exist, which is related to the order in which datapoints are selected for sampling households. This is shown in the example below:

Example:

Consider the situation of two overlapping circles in Figure 6.3.

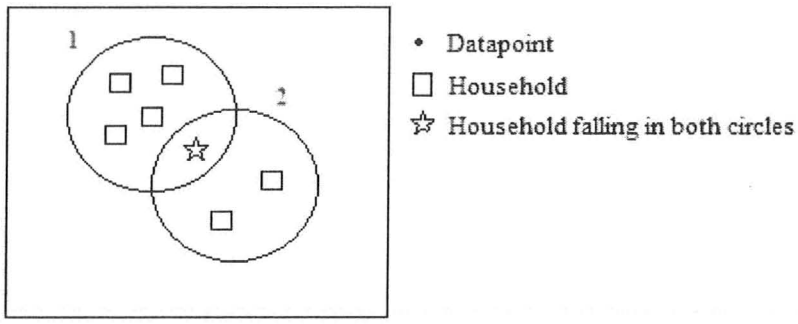


Figure 6.3: Case of two overlapping circles (circle 1 and circle 2).

If the first household sampled is from circle 1 (corresponding to datapoint 1) and the next household sampled is from circle 2 (corresponding to datapoint 2), the probabilities of selection for household □ or ☆ are:

Sample from datapoint 1: P(☆ or □) = 1/5
Sample[30] from datapoint 2: P(☆) = 1/3 and P(□) = 1/3
Sample[31] from datapoint 2: P(☆) = 0 and P(□) = 1/2

If the first household sampled is from circle 2 (corresponding to datapoint 2) and the next households sampled is from circle 1 (corresponding to datapoint 1), the probabilities of selection for household □ or ☆ are:

Sample from datapoint 2: P(☆ or □) = 1/3
Sample[32] from datapoint 1: P(☆) = 1/5 and P(□) = 1/5
Sample[33] from datapoint 1: P(□) = 0 and P(□) = 1/4

Hence, as per the probabilities obtained from the illustration above, 1 it matters which datapoint (DP) the first household is sampled from, and which datapoint the second household is sampled from (hence, all subsequent

---

30 If ☆ has not already been chosen in datapoint 1

31 If ☆ has already been chosen in datapoint 1

32 If ☆ has not already been chosen in datapoint 2

33 If ☆ has already been chosen in datapoint 2

households too) i.e. DP 1 then DP 2, or DP 1 then DP 2. However, it is also not clear as to which of the multiple probabilities of selection pertain most accurately to each household selected for sampling. Accordingly, a future extension may also look at this problem and see to what degree it affects estimates and standard errors. Also, in the simulations the area of the sampled datapoint circle is considered relevant when calculating the probability of selecting a household. But when overlapping is not ignored, one should consider how to calculate the area of the sampled datapoint circle when it is overlapping with a part of some other circle(s), and how it affects point estimates and standard errors.

A further extension would be to investigate the issue of different probabilities of selection for a household, depending on which datapoint was used to sample the household. Again, this might be a difficult issue to solve but one that would be very interesting given that this issue is inherent in the sampling method. In a situation where the town was made physically very large, e.g. 5000 * 5000 units, and the number of households in the town was increased to 10,000 households (but everything else kept the same), the issue of overlapping circles would then become very minor. However, the issue of different probabilities of selection would still exist because a specific household would still have a number of different probabilities for selection depending on which datapoint it got chosen through. So investigating this issue might give an idea as to how it affects parameter estimates.

Finally, another consideration has to do with the possible relationships between town characteristics. Right now, only basic, univariate characteristics have been looked at, but in the future bivariate relationships could be looked at too. These could be between binomial variables as shown in Figure 6.4, e.g. propose joint probabilities for gun ownership and confidence in police and see if there is any relationship between both variables. One would expect that if there is low confidence in police, then households may keep a gun for security reasons.

Gun O/S \ CIP	Low	High	
Y	0.25	0.05	
N	0.35	0.35	
	0.60	0.40	1

Figure 6.4: Example of a possible bivariate distribution of binomial variables.

Alternatively, the relationships could be between continuous variables, e.g. height and weight. In fact, the households in the town could be set up so that interactions among multivariate characteristics could also be investigated, instead of just bivariate characteristics. I have not done any work on any such future extensions and it might be complex to determine how to propose joint probabilities for the characteristics but reasonable estimates could be obtained from the available data.

D. Conclusion

After running the simulations and looking at the results, this new sampling method seems to provide estimates that although are not always exactly as expected, in many instances they are very close to what is expected e.g. the mean of samples mean for income and gun ownership, and expected standard error for income and gun ownership. The proportion of sample means falling within 1.96SE's of the true mean was a little above and below the expected 95% in both the standard and dense town and these differences are attributed to the sampling method. In summary, this sampling method does appear to be useful for purposes of sampling areas where gathering information about the population would otherwise be too difficult.

BIBLIOGRAPHY

Cluster Sampling. http://www.metagora.org/training/encyclopedia/cluster.html (accessed September 2, 2010).

Lethbridge, Russell. *NormRand - Produce random numbers with normal distribution.* November 25, 2000. http://www.devx.com/vb2themax/Tip/19233 (accessed June 3, 2010).

Madansky, Dr. Albert. *Weighted Standard Error.* http://www.analyticalgroup.com/download/WEIGHTED_MEAN.pdf (accessed July 30, 2010).

Sampling Techniques. http://www.geographyteachingtoday.org.uk/fieldwork/resource/fieldwork-techniques/sampling-techniques/ (accessed September 2, 2010).

APPENDICES

Appendix A: A Method to Calculate a Single Probability of Selection

Since a given household is picked from a circle of radius r, there are πr^2 datapoints around every household that can be used for selecting it. That means there are πr^2 circles that a given household can be picked from. In my simulations, since r = 20, this translates to roughly 1,256 circles ($\pi 20^2$). Each datapoint has a probability p_i associated with it, which is the probability of selecting a household from that circle depending on the number of households it contains.

Thus,

P(Any selection for sample chooses a given HH) = $\sum_{i=1}^{\pi r2} p(n_i)p(HH|i)$

where $p(n_i)$ = probability of selecting a datapoint i

and $p(HH|i)$ = probability of sampling a given household from datapoint i

Example:

Given a specific household (HH) can be picked from 1256 circles (for r = 20), assume that 1254 circles contain only the given HH, 1 circle contains two HHs (including the given HH) and 1 circle contains 5 HHs (including the given HH).

Then,

$p(n_i) = \frac{1}{1{,}000{,}000}$ for i = 1, … , 1256
$p(HH|i) = 1$ for i = 1, … , 1254
$p(HH|i) = \frac{1}{2}$ for i = 1255
$p(HH|i) = \frac{1}{5}$ for i = 1256

So,

P(Any selection for sample chooses a given HH)

$$= \sum_{i=1}^{1256} p(n_i)p(HH|i)$$
$$= \frac{1}{1{,}000{,}000}\left[\sum_{i=1}^{1254}(1) + \frac{1}{2} + \frac{1}{5}\right]$$
$$= \frac{1}{1{,}000{,}000}\left[1254 + \frac{1}{2} + \frac{1}{5}\right]$$
$$= \frac{1{,}254.70}{1{,}000{,}000}$$

Note that $\boldsymbol{p(n_i) = \frac{1}{1{,}000{,}000}}$***, since the standard town is of size 1,000,000 square units (1000 x 1000) and there can be a datapoint at each set of the coordinates***

So,

P(Any selection for sample does not choose a given HH)

$= 1 - \sum_{i=1}^{1256} p(n_i)p(HH|i)$

P(Any of the 100 selections for the sample does not choose this HH)[34]

$= \left[1 - \sum_{i=1}^{1256} p(n_i)p(HH|i)\right]^{100}$

So P(The given household is selected in the sample of 100 HHs)

$= 1\text{-}\left[1 - \sum_{i=1}^{1256} p(n_i)p(HH|i)\right]^{100}$

[34] Given that the sample size is n = 100

Appendix B: How the Sample was Drawn in the South Lebanon Study

I will briefly describe how the actual sample was chosen in the South Lebanon study. The population from which the sample was drawn included 144 towns with 312,209 people[35] and 3 other bigger towns.

When sampling from the population, probability proportional to estimated size (PPES) was used. The data available for sampling towns was obtained through voter rolls to estimate the number of people living in each town. The towns were treated as clusters.

First, 800 households were sampled from the three bigger towns in South Lebanon, including 400 households from Tyre, 200 from Marjayoun, and another 200 from Bint Jbeil. Then, a further 50 towns out of 144 other towns were sampled (excluding the three bigger towns), selecting 16 further households per town. Thus, another 800 households were sampled this way (i.e. 16 households/town * 50 towns). Therefore, in total 1,600 households were sampled (i.e. sample size = 1,600 households).

However, in reality only an approximate 1,400 households out of the targeted 1,600 households were able to provide an individual that could be interviewed for the study.

---

[35] The populations of each town were only estimates. So strictly speaking, *Probability Proportion to Estimated Size (PPES)* was used in the South Lebanon study.

Appendix C: Estimating Households with a Gun in the Town

In order to know the number of households who own a gun in the town, one needs to know how many circles have been sampled in total, including any empty circles with no buildings. The following example illustrates how to estimate the number of households owning a gun in the town.

Example:

In this scenario, eleven datapoints were sampled where eight contained no households, two contained two households and one contained one household, as shown in Figure B-1 below:

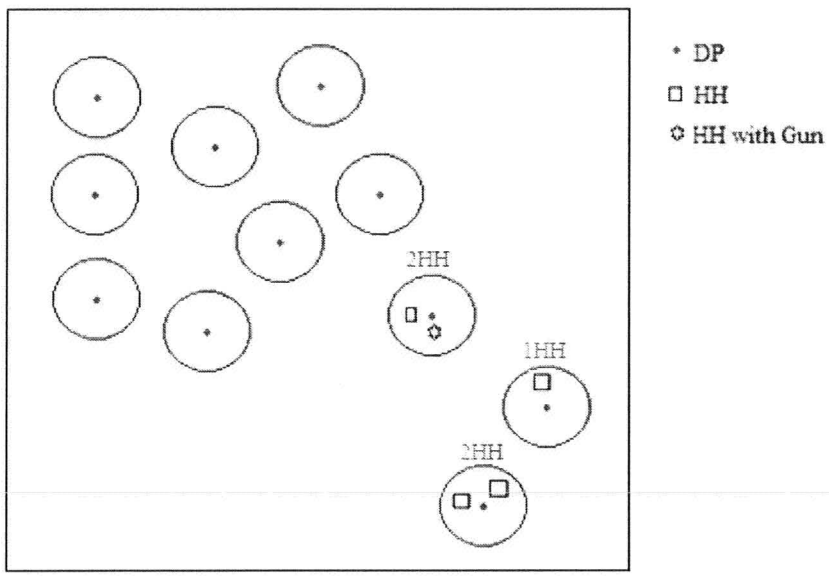


Figure B-1: Illustration of datapoints sampled in a town with eight containing no households and three containing some households.

Out of the three datapoints sampled with at least one household each, one of the datapoints with two households had a household owning a gun. Given the area of each datapoint circle was $\pi * 20^2$ and assuming no overlapping occurred between circles when the sample was drawn, the following calculations were done[36]:

[36] Moreover, it was also assumed that all circles lay inside the town, although in reality they could lie outside the town, in which case the area of those circles as a fraction of the total town area should be calculated first (using one of the five approaches in Appendix C) and then the calculations outlined below should be done, using the "adjusted area of the datapoint circle".

Calculations:

Fraction of town area sampled = $f_A = \frac{\text{Area (8 empty+3 non−empty circles) within town}}{\text{Area of Total Town}}$

Number of times town is larger than the town area sampled = $\frac{1}{f_A}$

Given,

w_i = number of households in sampled datapoint i[37], and
x_i = 0, 1 (0 - sampled household in the circle owns no gun;
1 - sampled household in the circle owns a gun)

Then,

w_i = 0 if there are no households in the sampled datapoint i

Now,

Estimate of households with a gun = $\sum w_i x_i$,
Proportion of households in the sample owning a gun = $\hat{p} = \frac{\sum w_i x_i}{\sum w_i}$
Estimate of total number of households with a gun in town = $\frac{\sum w_i x_i}{f_A}$

Hence,

$f_A = \frac{(11 * \pi * 20^2)}{1000*1000} = \frac{13{,}823}{1{,}000{,}000} = 0.013823$ of the total town area sampled

$\frac{1}{f_A} = \frac{1}{0.013823} = 72.3$ times town is larger than the town area sampled

$\sum w_i x_i = (2 * 1) + (2 * 0) + (1 * 0) = 2$ households are estimated to own a gun in the sampled town area

$\hat{p} = \frac{(2 * 1)+(2 * 0)+(1 * 0)}{2 + 2 + 1}$
$= \frac{2}{5}$ = the proportion of households in sample owning a gun

Thus,

$\frac{\sum w_i x_i}{f_A} = \sum w_i x_i * \frac{1}{f_A} = 2 * 72.3$
$= 144$ = the number of households estimated to own a gun in the town

[37] i = number of datapoints sampled with at least one household each

Appendix D: Five Approaches for Sampled Circle Area Calculations

There were five different approaches used when calculating the area of each sampled circle that was within the town. In this appendix, the conditions that determine which approach is to be used for each sampled circle have been identified. Moreover, graphical illustrations and the numerical work required to calculate the area for each approach, is also provided.

However, in order to determine which approach the datapoint followed, a few variables had to be defined first. These included:

T = Area of whole town in which the buildings lie = town length * town width
h = number of households in a datapoint circle
r = radius of circle
k = x-coordinate distance between the datapoint and town boundary
= min(x coordinate of datapoint, town length – x coordinate of datapoint)
j = y- coordinate distance between the datapoint and town boundary
= min(y coordinate of datapoint, town width – y coordinate of datapoint)
L = length between a and b
A = unshaded area of the circle, which is inside the town

Unshaded Area (A) Condition for all 5 Approaches:

1. If $j \geq 20$ and $k \geq 20$ → Use Approach 1
2. If $j \geq 20$ and $k < 20$ → Use Approach 2
3. If $j < 20$ and $k \geq 20$ → Use Approach 3
4. If $j^2 + k^2 < 400$, $j < 20$ and $k < 20$ → Use Approach 4
5. If $j^2 + k^2 > 400$, $j < 20$ and $k < 20$ → Use Approach 5

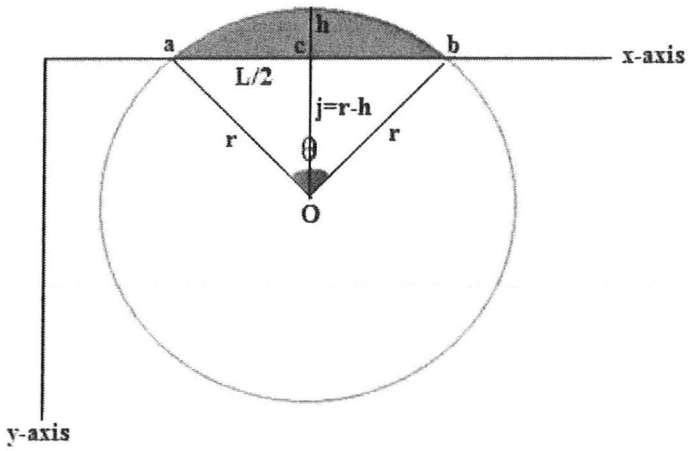
APPROACH 1:

In this approach, the whole circle is inside the town.

∴ Area of circle within the town = πr^2

APPROACH 2:

In this approach, part of the circle is outside the town on the x-axis. This can happen in two ways as shown in Figure C-1 below.



OR

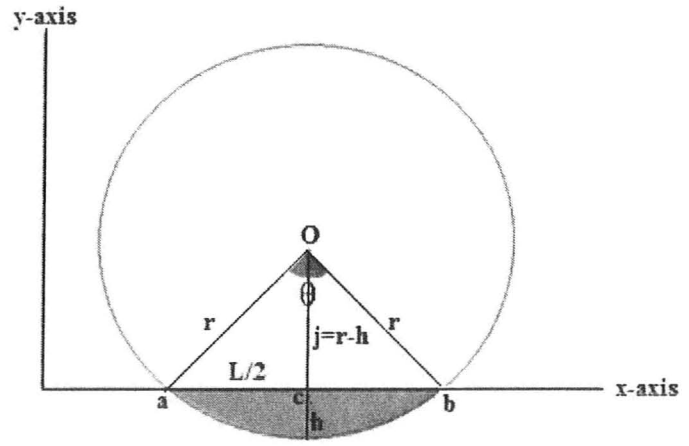


Figure C-1: In this scenario, part of the circle is outside the town on the x-axis.

Using, Radians ($360^o = 2\pi$) and h = r – j:

Area (circular segment) = Area (circular sector) – Area (isosceles triangle)

1. Area (circular sector)[38] $= \frac{1}{2}r^2\theta$

2. Area (isosceles triangle) = ½ × L × (r - h)

∴ Area (circular segment) = (½× r^2 × θ) – (½× L × (r - h))

To determine L & θ, using Pythagoras' Theorem:

1. $r^2 = (r - h)^2 + (L/2)^2$
   $L = \sqrt{8rh - 4h^2}$

2. $\sin(aOc) = \frac{\frac{L}{2}}{r}$, thus $aOc = \sin^{-1}(\frac{\frac{L}{2}}{r})$

   and, $\theta = 2 \times aOc$

   $$\boldsymbol{\theta = 2 \times \sin^{-1}(\frac{\frac{L}{2}}{r})}$$

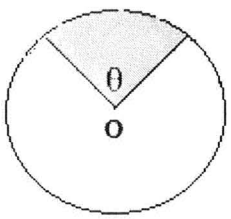
Solve using all variables:

Shaded Area of Circle[39] = (½× r^2 × θ) – (½× L × (r - h))

∴ Unshaded Area of Circle = A
= Area of Circle – Shaded Area of Circle
= πr^2 – [(½ × r^2 × θ) – (½× L × (r - h))]

---

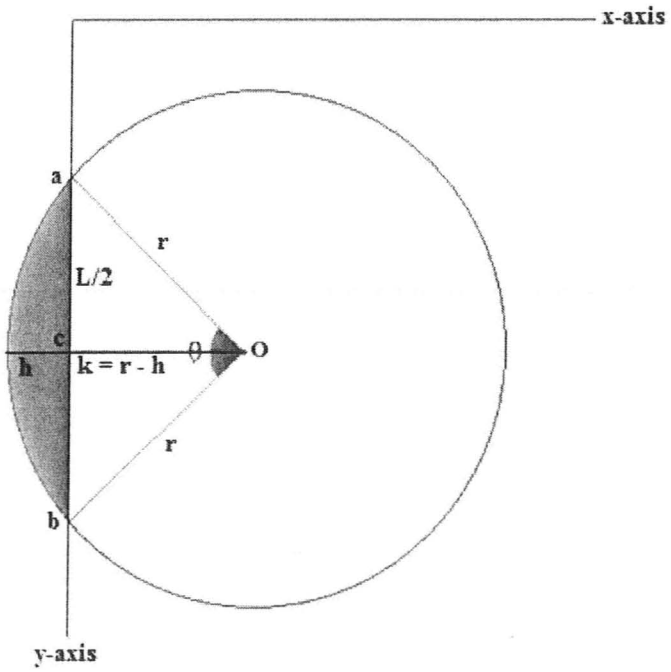
[38] Given the circle below, Area (circular sector) = $\pi r^2 \times \frac{\theta}{2\pi} = \frac{1}{2}r^2\theta$



[39] Shaded Area of Circle = Area (circular segment)

APPROACH 3:

In this approach, part of the circle is outside the town on the y-axis. This can happen in two ways as shown in Figure C-2 below.



OR

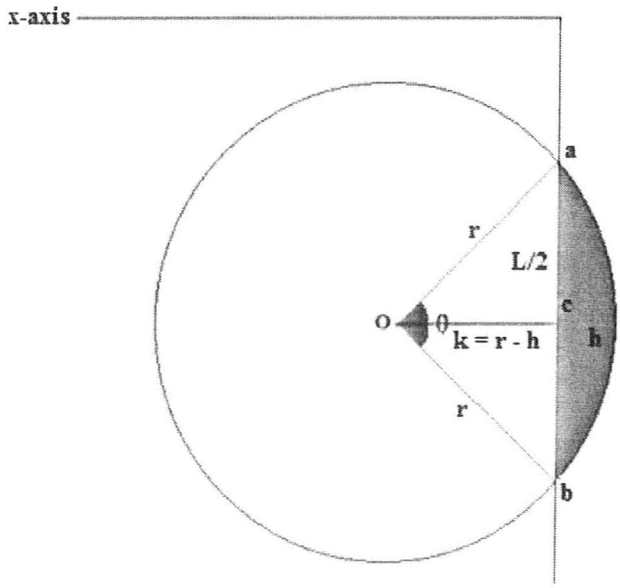


Figure C-2: In this scenario, part of the circle is outside the town on the y-axis.

In this approach, the same methodology is used as that used in Approach 2, thus,

Unshaded Area of Circle = A
= Area of Circle – Shaded Area of Circle
= $\pi r^2 - [(½ \times r^2 \times \theta) - (½ \times L \times (r - h))]$

APPROACH 4:

In this approach, the circle includes the edge of the town but part of the circle is outside the town (on both axes), as shown in Figure C-3 below. This can happen on any of the four edges of the town.

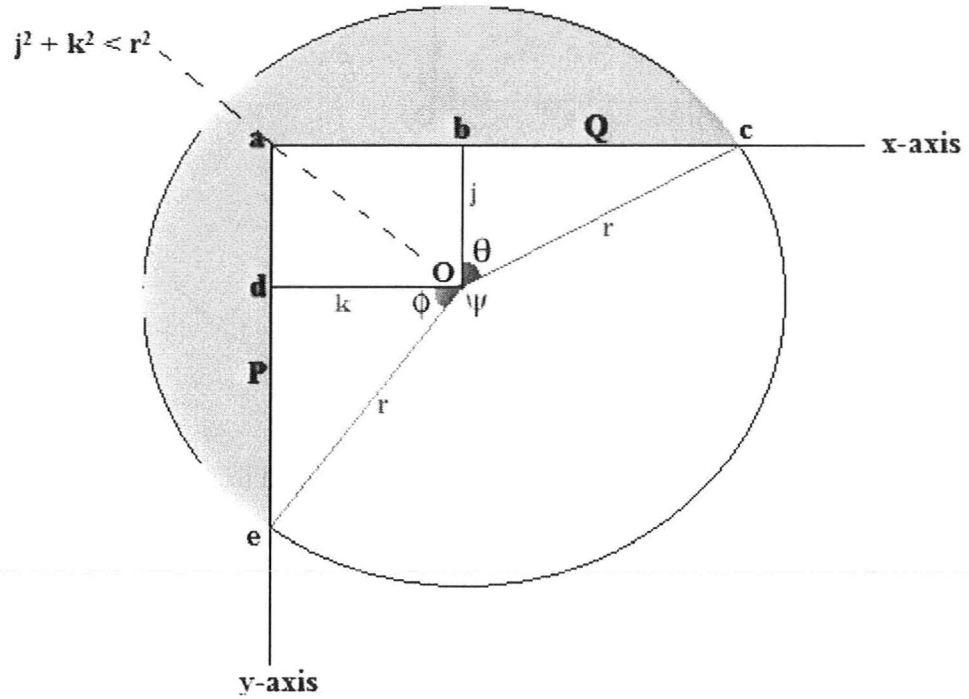


Figure C-3: In this scenario, the circle includes the corner but part of it is outside the town.

$j = \overline{ob} = \overline{ad}, Q = \overline{bc} = \sqrt{r^2 - j^2}, \theta = \sin^{-1}\frac{Q}{r}$

$k = \overline{od} = \overline{ab}, Q = \overline{de} = \sqrt{r^2 - k^2}, \varphi = \sin^{-1}\frac{P}{r}$

$\Psi = 2\pi - \frac{\pi}{2} - \theta - \varphi = \frac{3}{2}\pi - \theta - \varphi$

$\therefore$ Unshaded Area of Circle = A
$= jk + ½\, jQ + ½\, kP + ½\, r^2\Psi$

APPROACH 5:

In this approach, part of the circle is outside the town (on both axes) but the edge of the town is outside the circles, as shown in Figure C-4 below. This can happen on any of the four edges of the town.

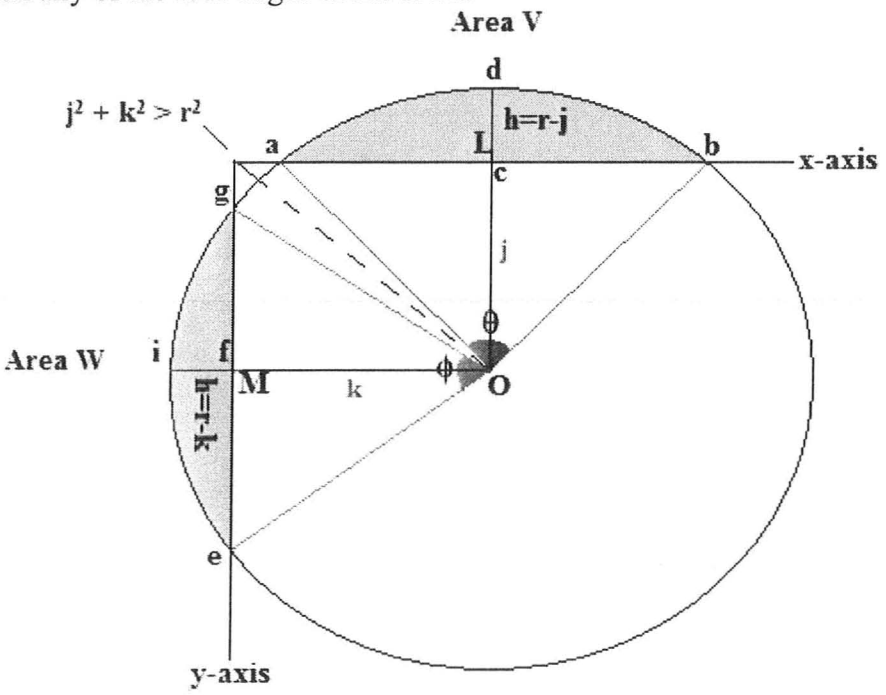


Figure C-4: In this scenario, part of the circle is outside the town (on both axes) but the edge of the town is outside the circles.

j = $\overline{oc}$, h = $\overline{cd}$ = r-j, L = $\overline{ab}$

$L = 2\sqrt{r^2 - j^2}$

$\theta = 2\sin^{-1}\frac{L}{2r}$

Area V $= \frac{1}{2}r^2\theta - \frac{jL}{2}$

k = $\overline{of}$, h = $\overline{if}$ = r-k, M = $\overline{eg}$

$M = 2\sqrt{r^2 - k^2}$

$\varphi = 2\sin^{-1}\frac{M}{2r}$

Area W $= \frac{1}{2}r^2\varphi - \frac{kM}{2}$

∴ **Unshaded Area of Circle = A**

$= \pi r^2 - V - W$

Appendix E: Overlapping Circles

In the simulations, it was assumed that sampling was done from datapoint circles that did not overlap with each other, as shown in Figure D-1.

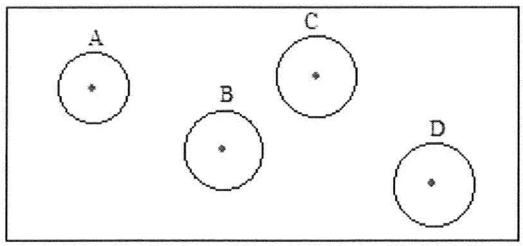


Figure D-1: Sampling households from circles which do not overlap. Thus, fraction of area sampled is $\frac{\text{Area(A)+ Area(B)+ Area(C)+ Area(D)}}{\text{Area of Total Town}}$.

In the simulations, a test was included in the code to check if the distance between any two datapoint centres in the sample was less than twice the radius. If that was true, then those two datapoints overlapped with each other and household(s) could lie in the overlap, as shown in Figure D-2.

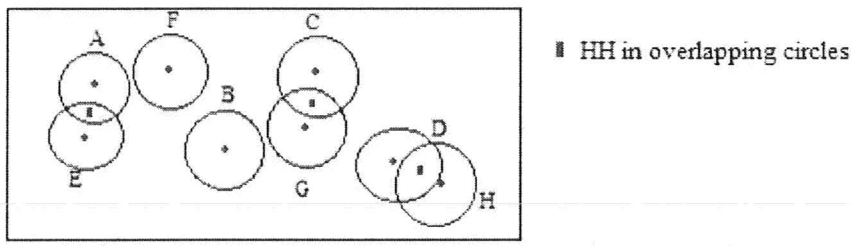


Figure D-2: Sampling households from circles that overlap, thus fraction of area sampled is

$$\frac{\text{[Area(A)+ Area (E)− Overlap] +Area (F)+ Area(B)+ [Area(C)+ Area(G)− Overlap]+ [Area(D)+ Area H−Overlap]}}{\text{Area of Total Town}}$$

As verified by simulations on both towns, overlapping occurs between some datapoint circles when collecting a sample. As expected, there is greater overlapping in the dense town compared to the standard town. Approximately 50% of circles in the dense town overlap with other circles in the sample and around 35% overlap in the standard town. In fact, in some selections of circles, more than two circles overlap with each other, which is also vouched by the simulations where there are times when four datapoints overlap with each other. Again, this occurrence is more common in the dense town, due to the way it is constructed. In such cases, there are many probabilities to consider when choosing

a household from an area of overlap. Thus, due to the complexity of these calculations, a simple case has been assumed where no circles overlap with each other, so that it is easier to calculate the weights and probabilities of selection. However, since this issue of overlapping circles was not built into the analysis, it is possible that some discrepancies between the true parameters estimates and those obtained from the samples may be due to households being chosen from circles that overlap with each other.

Overlapping Circles – Sampling without Replacement:

In the simulations, sampling was done without replacement, i.e. once a household was chosen, it could not be chosen again through a different datapoint. In order to illustrate the issue of overlapping and how it leads to multiple probabilities existing when selecting a household, I have provided an example below. Using sampling without replacement, I have outlined the different probabilities that are to be considered when choosing two different households from two overlapping circles, as shown in Figure D-3.

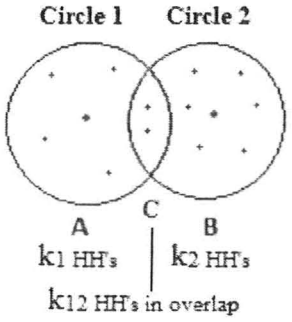


Figure D-3: Two overlapping circles, where the non-overlapping part of Circle 1 (A) contains k_1 households, the non-overlapping part of Circle 2 (B) contains k_2 households, and the overlapping part of both circles (C) contains k_{12} households.

The first household is chosen from DP 1 which corresponds to circle 1. The household could be picked from the non-overlap part of the circle A, or the overlap C. The second household is chosen from DP 2 which corresponds to circle 2. The household could be picked from the non-overlap part of the circle B, or the overlap C. The available choices for picking a household in each of the two circles are as shown in Figure D-4.

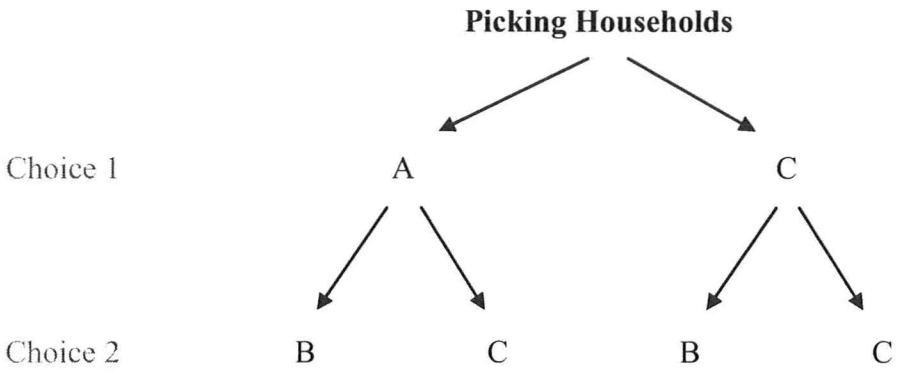


Figure D-4: Choices for picking the first household and the second household.

The probabilities for picking the first household through either A or C are shown in Figure D-5:

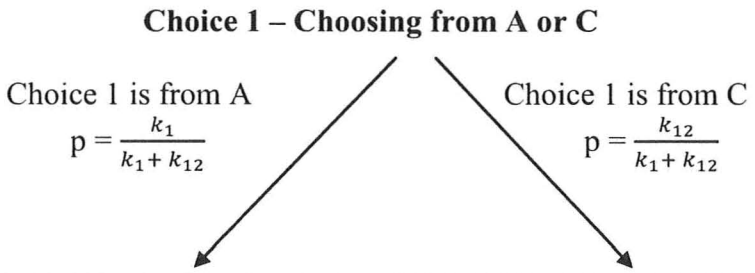


Figure D-5: Probabilities for picking the first household from DP 1.

And just as a check below, I have verified that P(1st HH is in A or C) = P(A) + P(C) = 1:

$$P(1^{st}\text{ HH is in A or C}) = P(\text{Household is in A}) + P(\text{Household is in C})$$
$$= \frac{k_1}{k_1 + k_{12}} + \frac{k_{12}}{k_1 + k_{12}} = 1$$

The probabilities for picking the second household from either B or C are conditional on whether the first household was picked from A or C, as shown in Figure D-6.

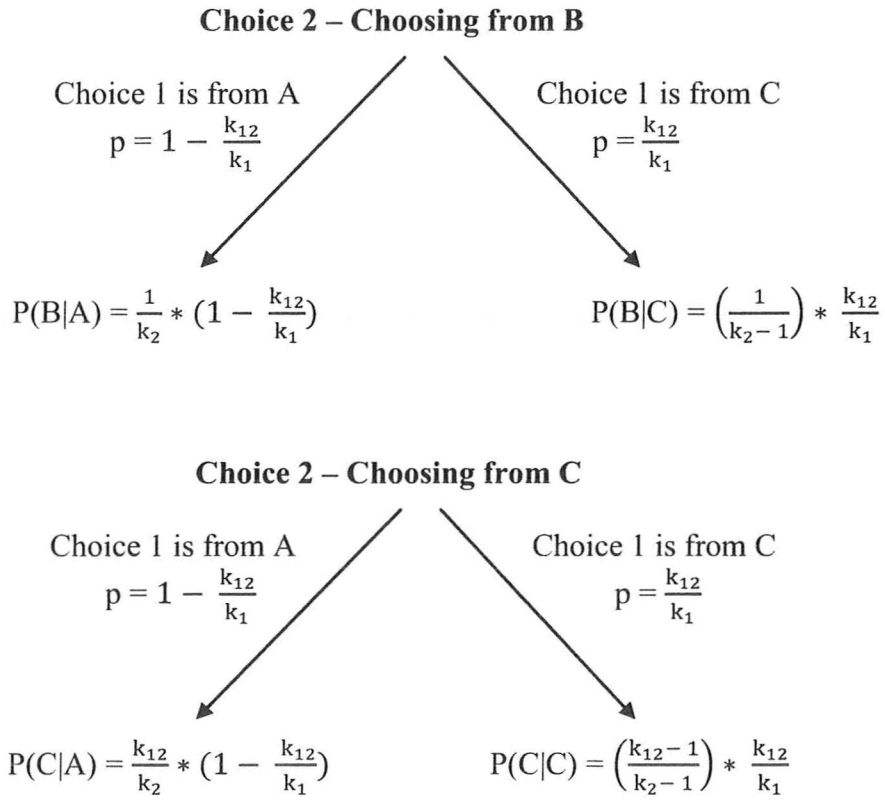


Figure D-6: Probabilities for picking the second household from DP 2.

And just as a check below, I have verified that P(2nd HH is in B or C) = P(B) + P(C) = 1:

$$P(B) = (k_2 - k_{12})\left[\left(\frac{k_{12}}{k_1}\right)\left(\frac{1}{k_2 - 1}\right) + \left(1 - \frac{k_{12}}{k_1}\right)\left(\frac{1}{k_2}\right)\right]$$
$$= (k_2 - k_{12})\left[\left(\frac{k_{12}}{k_1}\right)\left(\frac{1}{k_2 - 1}\right) + \left(\frac{k_1 - k_{12}}{k_1 k_2}\right)\right]$$

$$P(C) = (k_{12} - 1)\left[\left(\frac{k_{12}}{k_1}\right)\left(\frac{1}{k_2 - 1}\right) + (k_{12})\left[\left(1 - \frac{k_{12}}{k_1}\right)\left(\frac{1}{k_2}\right)\right]\right.$$
$$= (k_{12} - 1)\left[\left(\frac{k_{12}}{k_1}\right)\left(\frac{1}{k_2 - 1}\right) + (k_{12})\left[\left(\frac{k_1 - k_{12}}{k_1 k_2}\right)\right]\right.$$

$$\begin{aligned}
P(B) + P(C) &= (k_2 - k_{12})\left[\left(\frac{k_{12}}{k_1}\right)\left(\frac{1}{k_2 - 1}\right) + \left(\frac{k_1 - k_{12}}{k_1 k_2}\right)\right] \\
&+ (k_{12} - 1)\left[\left(\frac{k_{12}}{k_1}\right)\left(\frac{1}{k_2 - 1}\right) + (k_{12})\left[\left(\frac{k_1 - k_{12}}{k_1 k_2}\right)\right]\right. \\
&= (k_2 - k_{12})\left[\frac{\cancel{k_{12}k_2} + (k_1 k_2 - k_1) - (\cancel{k_2 k_{12}} - k_{12})}{k_1(k_2 - 1)k_2}\right] \\
&+ (k_{12} - 1)\left[\frac{k_2}{k_2} \cdot \frac{k_{12}}{k_1(k_2 - 1)}\right] + (k_{12})\left[\frac{k_2 - 1}{k_2 - 1} \cdot \frac{k_1 - k_{12}}{k_1 k_2}\right] \\
&= (k_2 - k_{12})\left[\frac{k_1 k_2 - k_1 + k_{12}}{k_1(k_2 - 1)k_2}\right] + (k_{12} - 1)\left[\frac{k_2 k_{12}}{k_1(k_2 - 1)k_2}\right] \\
&+ (k_{12})\left[\frac{k_2 k_1 - k_2 k_{12} - k_1 + k_{12}}{k_1(k_2 - 1)k_2}\right]
\end{aligned}$$

Given the denominator is the same $[k_1(k_2 - 1)k_2]$ for all three terms, let's just consider the numerator for all three terms:

$$\begin{aligned}
[P(B) + P(C)]_{Nu} &= (k_2 - k_{12})[k_1 k_2 - k_1 + k_{12}] + (k_{12} - 1)[k_2 k_{12}] \\
&+ (k_{12})[k_2 k_1 - k_2 k_{12} - k_1 + k_{12}] \\
&= k_2 k_1 k_2 - k_2 k_1 + \cancel{k_2 k_{12}} - \cancel{k_{12} k_1 k_2} + \cancel{k_{12} k_1} - \cancel{k_{12}^2} \\
&+ \cancel{k_{12} k_2 k_{12}} - \cancel{k_2 k_{12}} + \cancel{k_{12} k_2 k_1} - \cancel{k_{12} k_2 k_{12}} - \cancel{k_{12} k_1} + \cancel{k_{12}^2} \\
&= k_2 k_1 k_2 - k_2 k_1 \\
&= k_2 k_1 (k_2 - 1)
\end{aligned}$$

The final denominator and numerator are:

$$[P(B) + P(C)]_{Nu} = k_2 k_1 (k_2 - 1)$$
$$[P(B) + P(C)]_{De} = k_1(k_2 - 1)k_2$$

Hence,

➔ **P(2nd HH is in B or C)** = P(2nd HH is in B|1st HH is in A)
+ (2nd HH is in B|1st HH is in C)
+ (2nd HH is in C|1st HH is in A)
+ (2nd HH is in C|1st HH is in C)

$$= P(B) + P(C) = \frac{k_2 k_1 (k_2 - 1)}{k_1(k_2 - 1)k_2} = 1$$

In this example, I showed the different choices there are when picking a household from only two overlapping circles. However, if there are three, four or more overlapping circles, there are many more choices and probabilities for household selection, and the calculations end up become very complex and messy. For that reason, it was decided to keep the work simple and just assume that sampling was done from circles that did not overlap (so that there was only one probability to consider), even though the overlapping test in the code showed otherwise.

In practice, the amount of overlap depends on the size of the town and the number of datapoints, i.e. denser the town or larger the sample of datapoints collected, greater the expected overlapping.

Appendix F: Code Used for the Simulations

i. Building Module

This module was used to create the town, place buildings in it, allocate the "gun ownership" characteristics to some households, create the town graph and calculate the population parameters.

```
Sub building()

Application.ScreenUpdating = False
Application.DisplayAlerts = False
On Error Resume Next

' Specifying worksheets and starting row & column
OutputSheet = "Coordinates"
townGraph = "Town Graph"
StartRow = 15
StartColumn = 1

' Clears data & town graph to randomly generate new buildings coordinates each
time
Sheets(OutputSheet).Range(Cells(StartRow, 1), Cells(StartRow + 5000,
5)).ClearContents
Sheets(townGraph).Delete

' Matrix holding both x, y-coordinates of the 1000 buildings as they're created
TotalNumberOfBuildings = Sheets(OutputSheet).Cells(4, 2) +
Sheets(OutputSheet).Cells(4, 6)
ReDim buildingMatrix(TotalNumberOfBuildings, 2)

' Two iterations for High Income & Low Income
For IterationCount = 0 To 1

   Randomize
   ' Identifying inputs from the output sheet
   buildingNumber = Sheets(OutputSheet).Cells(4, 2 + 4 * IterationCount)
   buildingDistance = Sheets(OutputSheet).Cells(5, 2 + 4 * IterationCount)
   BoundaryDistance = Sheets(OutputSheet).Cells(6, 2 + 4 * IterationCount)
   TownLength = Sheets(OutputSheet).Cells(7, 2 + 4 * IterationCount)
   TownWidth = Sheets(OutputSheet).Cells(7, 4 + 4 * IterationCount)
   WestBoundaryDistance = Sheets(OutputSheet).Cells(8, 2 + 4 * IterationCount)
   gunPercentage = Sheets(OutputSheet).Cells(9, 2 + 4 * IterationCount)
```

```
    IncomeMean = Sheets(OutputSheet).Cells(10, 2 + 4 * IterationCount)
    IncomeStDev = Sqr(Sheets(OutputSheet).Cells(11, 2 + 4 * IterationCount))

    ' Keeps count of the number of buildings made in current iteration
    BuildingCount = 1

    ' Exit Criteria
    Do Until BuildingCount > buildingNumber

        IncorrectCoordinates = 0

        ' Generating random x,y coordinates for the current building being created

        xCoordinate = Int((TownLength + 1) * Rnd + 0)
        yCoordinate = Int((TownWidth + 1) * Rnd + 0)

        ' Verifies Coordinates
        If xCoordinate > TownLength Then IncorrectCoordinates = 1
        If yCoordinate > TownWidth Then IncorrectCoordinates = 1

        ' Checks coordinates against boundary walls
        If xCoordinate < BoundaryDistance Then IncorrectCoordinates = 1
        If xCoordinate < WestBoundaryDistance + BoundaryDistance Then
IncorrectCoordinates = 1
        If yCoordinate < BoundaryDistance Then IncorrectCoordinates = 1

        If xCoordinate > TownLength - BoundaryDistance Then
IncorrectCoordinates = 1
        If yCoordinate > TownWidth - BoundaryDistance Then IncorrectCoordinates
= 1

        ' Runs distance check for current building versus buildings already created,
since each building must be atleast "building distance' from all the other buildings
        If IncorrectCoordinates = 0 Then

            If IterationCount = 1 Then
                NumBuildingsToCheck = BuildingCount +
Sheets(OutputSheet).Cells(4, 2) - 1
            Else
                NumBuildingsToCheck = BuildingCount - 1
            End If

            For ThisBuilding = 1 To NumBuildingsToCheck
                hDistance = Abs(buildingMatrix(ThisBuilding, 1) - xCoordinate)
```

```
            vDistance = Abs(buildingMatrix(ThisBuilding, 2) - yCoordinate)
            DisBuild = Sqr(hDistance ^ 2 + vDistance ^ 2)

            If DisBuild < buildingDistance Then
               IncorrectCoordinates = 1
               Exit For
            End If
         Next
      End If

      ' Stores approved coordinates in a matrix and outputs the coordinates
      If IncorrectCoordinates = 0 Then
         buildingMatrix(BuildingCount + IterationCount *
Sheets(OutputSheet).Cells(4, 2), 1) = xCoordinate
         buildingMatrix(BuildingCount + IterationCount *
Sheets(OutputSheet).Cells(4, 2), 2) = yCoordinate

         ' Output building number, x and y-coordinate of each building being
created
         Sheets(OutputSheet).Cells(StartRow + BuildingCount - 1 + IterationCount
* Sheets(OutputSheet).Cells(4, 2), 1) = BuildingCount + IterationCount *
Sheets(OutputSheet).Cells(4, 2)
         Sheets(OutputSheet).Cells(StartRow + BuildingCount - 1 + IterationCount
* Sheets(OutputSheet).Cells(4, 2), 2) = xCoordinate
         Sheets(OutputSheet).Cells(StartRow + BuildingCount - 1 + IterationCount
* Sheets(OutputSheet).Cells(4, 2), 3) = yCoordinate

         ' Set Default Gun Ownership = 0
         Sheets(OutputSheet).Cells(StartRow + BuildingCount - 1 + IterationCount
* Sheets(OutputSheet).Cells(4, 2), 5) = 0

         ' Allocates income to buildings using a RGN from a Normal Distribution
(where income = Exp(R#) since ln Income is Normally distributed)
         Sheets(OutputSheet).Cells(StartRow + BuildingCount - 1 + IterationCount
* Sheets(OutputSheet).Cells(4, 2), 4) = Exp(NormRand[40] * IncomeStDev +
IncomeMean)
         Sheets(OutputSheet).Cells(StartRow + BuildingCount - 1 + IterationCount
* Sheets(OutputSheet).Cells(4, 2), 4).NumberFormat = "###,###"

         ' Increments building count as one more building is made and it's building
no, x, y-coordinate & income info is stored in matrix
         BuildingCount = BuildingCount + 1
```

[40] (Lethbridge 2000)

```
        End If

    Loop
    ' Do Loop Ends

    ' Allocates random buildings as gunOwners
    gunOwners = 0
    gunOwnership = Application.WorksheetFunction.RoundUp(gunPercentage *
buildingNumber, 0)
    Do Until gunOwners = gunOwnership

        'Initializes the variable for the first run and then re-initializes it every time
skipBuilding = 1 (upon satisfying the two 'If' conditions below)
        skipBuilding = 0

        ' Picks a random building
        ThisBuilding = Int((buildingNumber - 1 + 1) * Rnd + 1)

        ' If the building's been labelled as a gun owner, skip it and select another
building
        If Sheets(OutputSheet).Cells(StartRow + ThisBuilding - 1 + IterationCount *
Sheets(OutputSheet).Cells(4, 2), 5) = "1" Then skipBuilding = 1

        If skipBuilding = 0 Then

            Sheets(OutputSheet).Cells(StartRow + ThisBuilding - 1 + IterationCount *
Sheets(OutputSheet).Cells(4, 2), 5) = "1"

            gunOwners = gunOwners + 1

        End If

    Loop

Next
' Iteration Count Loop ends

'Bolds and centres font in column 1
Sheets(OutputSheet).Range(Cells(StartRow, StartColumn), Cells(StartRow +
10000, StartColumn)).Select
 Selection.Font.Bold = True
    With Selection
      .HorizontalAlignment = xlCenter
```

```
  End With

'Centres all font from columns 1-5
Sheets(OutputSheet).Range(Cells(StartRow, StartColumn + 1), Cells(StartRow +
10000, StartColumn + 4)).Select
Selection.Font.Bold = False
   With Selection
     .HorizontalAlignment = xlCenter
  End With

' Creates Town Graph
ActiveSheet.Shapes.AddChart.Select
ActiveChart.ChartType = xlXYScatter
ActiveChart.SeriesCollection(1).Name = townGraph
ActiveChart.SeriesCollection(1).XValues =
Sheets(OutputSheet).Range(Cells(StartRow, 2), Cells(StartRow +
Sheets(OutputSheet).Cells(4, 2) + Sheets(OutputSheet).Cells(4, 6) - 1, 2))
ActiveChart.SeriesCollection(1).Values =
Sheets(OutputSheet).Range(Cells(StartRow, 3), Cells(StartRow +
Sheets(OutputSheet).Cells(4, 2) + Sheets(OutputSheet).Cells(4, 6) - 1, 3))
For thisSeries = 2 To ActiveChart.SeriesCollection.Count
   ActiveChart.SeriesCollection(2).Delete
Next

TownLength = Sheets(OutputSheet).Cells(7, 2)
TownWidth = Sheets(OutputSheet).Cells(7, 4)

ActiveChart.Axes(xlCategory).Select
ActiveChart.Axes(xlCategory).MaximumScale = TownLength
ActiveChart.Axes(xlCategory).MinimumScale = 0
ActiveChart.Axes(xlCategory).HasTitle = True
ActiveChart.Axes(xlCategory).AxisTitle.Text = "EW Coordinates"
ActiveChart.Axes(xlValue).MaximumScale = TownWidth
ActiveChart.Axes(xlValue).MinimumScale = 0
ActiveChart.Axes(xlValue).HasTitle = True
ActiveChart.Axes(xlValue).AxisTitle.Text = "NS Coordinates"
ActiveChart.Legend.Delete

' Adds boundary line between the two income areas
ActiveChart.SeriesCollection.NewSeries
ActiveChart.SeriesCollection(2).XValues = Sheets(OutputSheet).Range(Cells(8,
3), Cells(8, 4))
ActiveChart.SeriesCollection(2).Values = Sheets(OutputSheet).Range(Cells(6, 4),
Cells(7, 4))
```

```
ActiveChart.SeriesCollection(2).ChartType = xlXYScatterSmoothNoMarkers
ActiveChart.SeriesCollection(2).Border.Color = RGB(155, 187, 89)
ActiveChart.SeriesCollection(2).Name = "Boundary Line"
ActiveChart.Location Where:=xlLocationAsNewSheet, Name:=townGraph
Sheets(townGraph).Move After:=Sheets(Sheets.Count)

' Colours Gun Owner Buildings
For ThisBuilding = 1 To Sheets(OutputSheet).Cells(4, 2) +
Sheets(OutputSheet).Cells(4, 6)
   If Sheets(OutputSheet).Cells(StartRow + ThisBuilding - 1, 5) = "1" Then
      With ActiveChart.SeriesCollection(1).Points(ThisBuilding)
         .MarkerBackgroundColor = RGB(255, 255, 255)
         .MarkerForegroundColor = RGB(255, 0, 0)
         .MarkerStyle = 8
         .MarkerSize = 7
         .Format.Line.Weight = 2.25

      End With
   End If
Next

ActiveChart.SeriesCollection(1).Border.LineStyle = xlLineStyleNone

Sheets(OutputSheet).Activate

'Set Range for Building Output
EndRow = StartRow + TotalNumberOfBuildings - 1
Set BuildingsInTown = Range(Cells(StartRow, StartColumn), Cells(EndRow,
StartColumn + 4))
ActiveWorkbook.Names.Add Name:="BuildingsInTown",
RefersTo:=BuildingsInTown

' Calculate Mean and SD for All Income
IncomeCol = 4

Set IncomeRange = Range(Cells(StartRow, IncomeCol), Cells(EndRow,
IncomeCol))
ActiveWorkbook.Names.Add Name:="IncomeRange", RefersTo:=IncomeRange

Range("IncomeMean").FormulaR1C1 = "=AVERAGE(IncomeRange)"
Range("IncomeStdDev").FormulaR1C1 = "=Stdev(IncomeRange)"

' Calculate Mean and SD for Low Income
EndRowLow = StartRow + Sheets(OutputSheet).Cells(4, 2) - 1
```

```
Set IncomeRangeLow = Range(Cells(StartRow, IncomeCol), Cells(EndRowLow,
IncomeCol))
ActiveWorkbook.Names.Add Name:="IncomeRangeLow",
RefersTo:=IncomeRangeLow

Range("LowIncomeMean").FormulaR1C1 = "=AVERAGE(IncomeRangeLow)"
Range("LowIncomeStdDev").FormulaR1C1 = "=Stdev(IncomeRangeLow)"

' Calculate Mean and SD for High Income
StartRowHigh = EndRowLow + 1

Set IncomeRangeHigh = Range(Cells(StartRowHigh, IncomeCol),
Cells(EndRow, IncomeCol))
ActiveWorkbook.Names.Add Name:="IncomeRangeHigh",
RefersTo:=IncomeRangeHigh

Range("HighIncomeMean").FormulaR1C1 = "=AVERAGE(IncomeRangeHigh)"
Range("HighIncomeStdDev").FormulaR1C1 = "=Stdev(IncomeRangeHigh)"

' Last Instructions for the Sub
Sheets(OutputSheet).Activate
Sheets(OutputSheet).Range("A1").Select

Application.ScreenUpdating = True
Application.DisplayAlerts = True
On Error GoTo 0

End Sub


Function NormRand() As Double
  ' NormRand returns a randomly distributed drawing from a
  ' standard normal distribution i.e. one with:
  ' Average = 0 and Standard Deviation = 1.0
  Dim fac As Double, rsq As Double, v1 As Double, v2 As Double
  Static flag As Boolean, gset As Double

  ' Each pass through the calculation of the routine produces
  '  two normally-distributed deviates, so we only need to do
  '  the calculations every other call. So we set the flag
  '  variable (to true) if gset contains a spare NormRand value.
  If flag Then
    NormRand = gset
```

```
    ' Force calculation next time.
    flag = False
  Else
    ' Don't have anything saved so need to find a pair of values
    ' First generate a co-ordinate pair within the unit circle:
    Do
      v1 = 2 * Rnd - 1#
      v2 = 2 * Rnd - 1#
      rsq = v1 * v1 + v2 * v2
    Loop Until rsq <= 1#

    ' Do the Math:
    fac = Sqr(-2# * Log(rsq) / rsq)

    ' Return one of the values and save the other (gset) for next time:
    NormRand = v2 * fac
    gset = v1 * fac
    flag = True
  End If

End Function
```

ii. Sample Module

This code was used to create a sample of n households, calculate the sample parameters for income and gun ownership using weights and probabilities, and also calculate estimates for a set of samples drawn from the population.

```
' Initializing Public Variables (to be used across multiple modules)
Option Explicit
Public IterationCount As Double
Public ThisIteration As Double
Public sngElapsed As Long
Public TimeRemaining As Long
Public IterationCountStr As String
Public ShowFormFlag As Integer


' Starting Sub
Public Sub sample()

Dim PasteToCol As Double
PasteToCol = 1
```

```
'Number of iterations required by user (each iteration = a sample of 'Sample
Count' observations)
IterationCount = Int(IterationCountStr)
If IterationCount < 1 Then Exit Sub
If str(IterationCount) = "" Then Exit Sub

Range("ItrCount").Value = IterationCount

Application.ScreenUpdating = False
Application.DisplayAlerts = False
On Error Resume Next

' Initializing Variables
Dim OutputSheet As String
Dim IterationSheet As String
Dim InputSheet As String
Dim StartRow As Double
Dim StartColumn As Double
Dim IterationOutputRow As Double
Dim RequiredSample As Double
Dim BuildingRadius As Double
Dim BoundaryDistance As Double
Dim TownLength As Double
Dim TownWidth As Double
Dim TownArea As Double
Dim WestBoundaryDistance As Double
Dim PlacedBuildings As Double

Dim SampleCount As Double
Dim dpCount As Double
Dim HighCount As Double
Dim LowCount As Double
Dim BuildingCount As Double
Dim IncorrectCoordinates As Double
Dim NumberBuildingsDone As Double
Dim xCoordinate As Double
Dim yCoordinate As Double
Dim ThisBuilding As Double
Dim hDistance As Double
Dim vDistance As Double
Dim Distance As Double
Dim CurrentBuilding As Double
Dim TotalWeight1 As Double
```

```
Dim TotalWeight2 As Double
Dim MeanWeight1 As Double
Dim MeanWeight2 As Double
Dim ThisRow As Double
Dim LoopStartRow As Double
Dim LoopEndRow As Double
Dim stdDevNum(2, 2) As Double
Dim stdDevDen(2) As Double
Dim ThisCase As Double
Dim a As Double
Dim CurrentRow As Double
Dim hDistanceDP As Double
Dim vDistanceDP As Double
Dim IterationSum As Double
Dim Avg As Double
Dim IterationNo As Double
Dim StdDev As Double
Dim PctDone As Double

Dim ProbabilityChooseHouseHold_1 As Double
Dim WeightofHouseholdChosen_1 As Double
Dim ProbabilityChooseHouseHold_2 As Double
Dim WeightofHouseholdChosen_2 As Double

Dim XDistFromNearestBorder As Double
Dim YDistFromNearestBorder As Double
Dim RawAreaAroundDataPoint As Double
Dim SelectApproach As Double
Dim AreaofCircle As Double
Dim EffectiveArea As Double
Dim LengthofRadiusOutsideTown_Y As Double
Dim LengthofRadiusOutsideTown_X As Double
Dim ChordLength_Y As Double
Dim ChordLength_X As Double
Dim AngleTheta As Double
Dim AnglePhi As Double
Dim AnglePsi As Double
Dim ThetaOpposite As Double
Dim PhiOpposite As Double
Dim EffectiveArea_X As Double
Dim EffectiveArea_Y As Double

Dim sngStart As Double, sngEnd As Double
Dim ValueofPi As Double
```

```
ValueofPi = Application.WorksheetFunction.Pi()

Dim OutputSheetOutputRow As Integer
Dim OutputSheetEndRow As Integer
Dim OutputSheetOutputCol As Integer
Dim Case1IncomeAverageCol As Integer
Dim Case1IncomeSErrorCol As Integer
Dim Case1GunOwnershipAverageCol As Integer
Dim Case1GunOwnershipSErrorCol As Integer
Dim Case2IncomeAverageCol As Integer
Dim Case2IncomeSErrorCol As Integer
Dim Case2GunOwnershipAverageCol As Integer
Dim Case2GunOwnershipSErrorCol As Integer
Dim Case1IncomeAverageRange As Range
Dim Case1IncomeSErrorRange As Range
Dim Case1GunOwnershipAverageRange As Range
Dim Case1GunOwnershipSErrorRange As Range
Dim Case2IncomeAverageRange As Range
Dim Case2IncomeSErrorRange As Range
Dim Case2GunOwnershipAverageRange As Range
Dim Case2GunOwnershipSErrorRange As Range

Dim Case1IncomeSErrorSqCol As Integer
Dim Case1GunOwnershipSErrorSqCol As Integer
Dim Case1IncomeSErrorSqRange As Range
Dim Case1GunOwnershipSErrorSqRange As Range

Dim EndRow As Integer
Dim StartingColumn As Integer
Dim BuildingNumberRange As Range
Dim BuildingXCoordinate As Range
Dim BuildingIncome As Range

StartingColumn = 26
sngElapsed = 0

' Clears data outputted on the Statistical Output sheet for every new set of
iterations
OutputSheet = "Statistical Output"
Sheets(OutputSheet).Select
Sheets(OutputSheet).Rows("5:100000").Clear

' Clears data outputted on the Iteration Output sheet for every new set of iterations
```

```
IterationSheet = "Iteration Output"
Sheets(IterationSheet).Select
Sheets(IterationSheet).Rows("4:1000000").Clear

' Specifying input sheet and identifying inputs from the input sheet (input sheet in
Mod 2 = output sheet from Mod 1)
InputSheet = "Coordinates"
StartRow = 15
StartColumn = 10
IterationOutputRow = 4

RequiredSample = Sheets(InputSheet).Cells(4, StartColumn + 1)
BuildingRadius = Sheets(InputSheet).Cells(5, StartColumn + 1)
BoundaryDistance = Sheets(InputSheet).Cells(6, StartColumn + 1)
TownLength = Sheets(InputSheet).Cells(7, StartColumn + 1)
TownWidth = Sheets(InputSheet).Cells(7, StartColumn + 3)
TownArea = TownLength * TownWidth
WestBoundaryDistance = Sheets(InputSheet).Cells(8, StartColumn + 1)
PlacedBuildings = Sheets(InputSheet).Cells(4, StartColumn - 8) +
Sheets(InputSheet).Cells(4, StartColumn - 4)

' Running the sample of 'Sample Count' observations a specified number of
iterations and storing the statistical output for each iteration on the Statistical
Output sheet
For ThisIteration = 1 To IterationCount
   Randomize
   sngStart = Timer

   Sheets(InputSheet).Select

   ' Clears sample data to randomly generate new buildings coordinates for each
iteration
   Sheets(InputSheet).Range(Cells(StartRow, StartColumn), Cells(StartRow +
5000, 24)).ClearContents

   SampleCount = 0
   dpCount = 1
   HighCount = 0
   LowCount = 0

   ' Stores the numerator for the Standard Deviation numerator BY CASE (rows)
and income/gun ownership (columns) in a matrix
   'Dim stdDevNum(2, 2)
```

```
' Stores the denominator for the Standard Deviation denominator BY CASE (independant of income & gun ownership values) in a matrix
'Dim stdDevDen(2)

' Stores information of buildings already chosen in previous datapoints (marked as '1') when creating the sample (array = max number of buildings in town), where placedBuildings = total buildings in town
ReDim buildingDone(PlacedBuildings)

' Exit criteria for the sample which equals a specified number of households (THIS set of code will be a run a specified number of iterations)
Do Until SampleCount = RequiredSample

    BuildingCount = 0
    IncorrectCoordinates = 0
    NumberBuildingsDone = 0

    ' Generating random coordinates for a DP randomly selected
    xCoordinate = Int((TownLength + 1) * Rnd + 0)
    yCoordinate = Int((TownWidth + 1) * Rnd + 0)

    If xCoordinate > TownLength Then IncorrectCoordinates = 1
    If yCoordinate > TownWidth Then IncorrectCoordinates = 1

    ' Checks coordinates against boundary walls
    If xCoordinate < BoundaryDistance Then IncorrectCoordinates = 1
    If yCoordinate < BoundaryDistance Then IncorrectCoordinates = 1

    If xCoordinate > TownLength - BoundaryDistance Then IncorrectCoordinates = 1
    If yCoordinate > TownWidth - BoundaryDistance Then IncorrectCoordinates = 1

    ' Outputs the coordinates for the specified DP
    If IncorrectCoordinates = 0 Then

        ' Cleared for every new random DP selected; Holds info of all buildings lying within the DP including the building number from the original buildingMatrix, income and gun ownership value (array for each DP = max number of buildings in town)
        ReDim buildingsInRadius(PlacedBuildings, 3)
```

```
        ' Runs distance check for all buildings in town to check whether any are
within the DP's radius and if they are, stores their building number, income and
gun o/w in the array
        For ThisBuilding = 1 To PlacedBuildings

            hDistance = Abs(Sheets(InputSheet).Cells(StartRow + ThisBuilding - 1,
StartColumn - 8) - xCoordinate)
            vDistance = Abs(Sheets(InputSheet).Cells(StartRow + ThisBuilding - 1,
StartColumn - 7) - yCoordinate)
            Distance = Sqr(hDistance ^ 2 + vDistance ^ 2)

            If Distance < BuildingRadius Then
               BuildingCount = BuildingCount + 1
               buildingsInRadius(BuildingCount, 1) = ThisBuilding
               buildingsInRadius(BuildingCount, 2) =
Sheets(InputSheet).Cells(StartRow + ThisBuilding - 1, StartColumn - 6)
               buildingsInRadius(BuildingCount, 3) =
Sheets(InputSheet).Cells(StartRow + ThisBuilding - 1, StartColumn - 5)
            End If

        Next

        ' Outputs the DP number being sampled and bolds & centres it
        Sheets(InputSheet).Cells(StartRow + dpCount - 1, StartColumn) =
dpCount
        Sheets(InputSheet).Cells(StartRow + dpCount - 1,
StartColumn).Font.Bold = True
        Sheets(InputSheet).Cells(StartRow + dpCount - 1,
StartColumn).HorizontalAlignment = xlCenter

        ' Outputs the randomly generated x & y-coordinate of the DP being
sampled
        Sheets(InputSheet).Cells(StartRow + dpCount - 1, StartColumn + 2) =
xCoordinate
        Sheets(InputSheet).Cells(StartRow + dpCount - 1, StartColumn + 3) =
yCoordinate

        ' By default, all coordinates are listed as "Low"
        Sheets(InputSheet).Cells(StartRow + dpCount - 1, StartColumn + 4) =
"Low"
        LowCount = LowCount + 1

        ' Checks coordinate to verify "Low" or "High"
        If xCoordinate <= WestBoundaryDistance Then
```

```
            Sheets(InputSheet).Cells(StartRow + dpCount - 1, StartColumn + 4) =
"High"
            HighCount = HighCount + 1
            LowCount = LowCount - 1
        End If

        ' Outputs number of buildings within the radius, where 'buildingCount'
keeps a count of all buildings lying within the specific DP radius
        Sheets(InputSheet).Cells(StartRow + dpCount - 1, StartColumn + 1) =
BuildingCount

        ' Counts how many of the buildings in the radius have been selected in a
previous DP, storing in a variable called numberBuildingsDone
        For CurrentBuilding = 1 To BuildingCount

            'Gets the building number of all buildings in the DP (stored in the
buildingsInRadius matrix) and checks whether those buildings have already been
chosen ('1') by checking the info through the buildingDone matrix
            If buildingDone(buildingsInRadius(CurrentBuilding, 1)) = 1 Then
                NumberBuildingsDone = NumberBuildingsDone + 1
            End If
        Next

        ' If all the buildings in the radius haven't been previously selected, then a
random building is selected
        If BuildingCount > 0 Then
            If NumberBuildingsDone < BuildingCount Then
                CurrentBuilding = Int((BuildingCount - 1 + 1) * Rnd + 1)

                'Above statement selects a random building; below Do loop checks
that if it's not already chosen (by verifying through the buildingDone matrix) then
it'll be selected, else buildings will continue to be randomly chosen until one is
picked which hasn't been previously selected
                Do Until buildingDone(buildingsInRadius(CurrentBuilding, 1)) = 0
                    CurrentBuilding = Int((BuildingCount - 1 + 1) * Rnd + 1)
                Loop

                ' Just verifies again that the building chosen hasn`t already been
selected before
                If buildingDone(buildingsInRadius(CurrentBuilding, 1)) = 0 Then

                    'Outputs the building number, income & gun o/s value from the
info stored in the buildingsInRadius Matrix for that specific building
```

```
                Sheets(InputSheet).Cells(StartRow + dpCount - 1, StartColumn +
5) = buildingsInRadius(CurrentBuilding, 1)
                Sheets(InputSheet).Cells(StartRow + dpCount - 1, StartColumn +
6) = buildingsInRadius(CurrentBuilding, 2)
                Sheets(InputSheet).Cells(StartRow + dpCount - 1, StartColumn +
7) = buildingsInRadius(CurrentBuilding, 3)

                buildingDone(buildingsInRadius(CurrentBuilding, 1)) = 1

                ' Increments count for an additional building sampled
                SampleCount = SampleCount + 1
              End If
            End If
          End If

        ' CALCULATION OF WEIGHTS & PROBABILITIES USING THE 5
APPROACHES

        XDistFromNearestBorder = Application.Min(xCoordinate, TownLength -
xCoordinate)
        YDistFromNearestBorder = Application.Min(yCoordinate, TownWidth -
yCoordinate)
        RawAreaAroundDataPoint = XDistFromNearestBorder ^ 2 +
YDistFromNearestBorder ^ 2

        'Given the above distances, select appropriate approach to calculate area
around selected data point that is within the town
        If XDistFromNearestBorder >= BuildingRadius And
YDistFromNearestBorder >= BuildingRadius Then
            SelectApproach = 1
            ElseIf XDistFromNearestBorder >= BuildingRadius And
YDistFromNearestBorder < BuildingRadius Then
              SelectApproach = 2
            ElseIf XDistFromNearestBorder < BuildingRadius And
YDistFromNearestBorder >= BuildingRadius Then
              SelectApproach = 3
            ElseIf RawAreaAroundDataPoint <= BuildingRadius ^ 2 Then
              SelectApproach = 4
            ElseIf RawAreaAroundDataPoint > BuildingRadius ^ 2 And
XDistFromNearestBorder < BuildingRadius And YDistFromNearestBorder <
BuildingRadius Then
              SelectApproach = 5
            Else
              MsgBox ("Selection Invalid - Select Approach Again")
```

```
        End If

        ' Calculating area of a whole circle
        AreaofCircle = BuildingRadius * BuildingRadius * ValueofPi

        ' Defining the "area of circle within the town" using Approaches 1-5
        Select Case SelectApproach

          Case 1
            EffectiveArea = AreaofCircle

          Case 2
            LengthofRadiusOutsideTown_Y = BuildingRadius -
YDistFromNearestBorder
            ChordLength_Y = Sqr(8 * BuildingRadius *
LengthofRadiusOutsideTown_Y - 4 * LengthofRadiusOutsideTown_Y ^ 2)
            AngleTheta = 2 *
Application.WorksheetFunction.Asin(ChordLength_Y / (2 * BuildingRadius))
            EffectiveArea = AreaofCircle - (1 / 2 * (BuildingRadius ^ 2) *
AngleTheta - 1 / 2 * ChordLength_Y * YDistFromNearestBorder)

          Case 3
            LengthofRadiusOutsideTown_X = BuildingRadius -
XDistFromNearestBorder
            ChordLength_X = Sqr(8 * BuildingRadius *
LengthofRadiusOutsideTown_X - 4 * LengthofRadiusOutsideTown_X ^ 2)
            AngleTheta = 2 *
Application.WorksheetFunction.Asin(ChordLength_X / (2 * BuildingRadius))
            EffectiveArea = AreaofCircle - (1 / 2 * (BuildingRadius ^ 2) *
AngleTheta - 1 / 2 * ChordLength_X * XDistFromNearestBorder)

          Case 4
            ThetaOpposite = Sqr(BuildingRadius ^ 2 - YDistFromNearestBorder
^ 2)
            PhiOpposite = Sqr(BuildingRadius ^ 2 - XDistFromNearestBorder ^
2)
            AngleTheta = Application.WorksheetFunction.Asin(ThetaOpposite /
BuildingRadius)
            AnglePhi = Application.WorksheetFunction.Asin(PhiOpposite /
BuildingRadius)
            AnglePsi = 3 / 2 * ValueofPi - AngleTheta - AnglePhi
            EffectiveArea = (XDistFromNearestBorder *
YDistFromNearestBorder) + (1 / 2 * PhiOpposite * XDistFromNearestBorder) +
```

```
(1 / 2 * ThetaOpposite * YDistFromNearestBorder) + (1 / 2 * BuildingRadius ^ 2
* AnglePsi)

            Case 5
              LengthofRadiusOutsideTown_Y = BuildingRadius -
YDistFromNearestBorder
              LengthofRadiusOutsideTown_X = BuildingRadius -
XDistFromNearestBorder
              ChordLength_Y = Sqr(8 * BuildingRadius *
LengthofRadiusOutsideTown_Y - 4 * LengthofRadiusOutsideTown_Y ^ 2)
              ChordLength_X = Sqr(8 * BuildingRadius *
LengthofRadiusOutsideTown_X - 4 * LengthofRadiusOutsideTown_X ^ 2)
              AngleTheta = 2 *
Application.WorksheetFunction.Asin(ChordLength_Y / (2 * BuildingRadius))
              AnglePhi = 2 *
Application.WorksheetFunction.Asin(ChordLength_X / (2 * BuildingRadius))

              EffectiveArea_Y = (1 / 2 * (BuildingRadius ^ 2) * AngleTheta - 1 / 2
* ChordLength_Y * YDistFromNearestBorder)
              EffectiveArea_X = (1 / 2 * (BuildingRadius ^ 2) * AnglePhi - 1 / 2 *
ChordLength_X * XDistFromNearestBorder)
              EffectiveArea = AreaofCircle - EffectiveArea_Y - EffectiveArea_X
          End Select

          ' Calculates the Probability and Weight for Case 1
          If BuildingCount > 0 Then
            If BuildingCount - NumberBuildingsDone > 0 Then
              ProbabilityChooseHouseHold_1 = (EffectiveArea / TownArea) * (1 /
(BuildingCount - NumberBuildingsDone))
              WeightofHouseholdChosen_1 = 1 / ProbabilityChooseHouseHold_1

              Sheets(InputSheet).Cells(StartRow + dpCount - 1, StartColumn + 8) =
ProbabilityChooseHouseHold_1
              Sheets(InputSheet).Cells(StartRow + dpCount - 1, StartColumn + 9) =
WeightofHouseholdChosen_1

            End If
          End If

          ' Calculates the Probability and Weight for Case 2
          If BuildingCount > 0 Then
              ProbabilityChooseHouseHold_2 = (EffectiveArea / TownArea) * (1 /
BuildingCount)
              WeightofHouseholdChosen_2 = 1 / ProbabilityChooseHouseHold_2
```

```
            Sheets(InputSheet).Cells(StartRow + dpCount - 1, StartColumn + 10)
= ProbabilityChooseHouseHold_2
            Sheets(InputSheet).Cells(StartRow + dpCount - 1, StartColumn + 11)
= WeightofHouseholdChosen_2
        End If

        ' Increments count for an additional DP sampled (whether or not it
contained a building that was sampled)
        dpCount = dpCount + 1

      End If

   Loop

   ' Calculates and outputs mean weight for both case 1 and case 2

   TotalWeight1 = 0
   TotalWeight2 = 0
   MeanWeight1 = 0
   MeanWeight2 = 0
   EndRow = StartRow + dpCount - 2 'Additional -1 due to dp count + 1 at end of
loop


   For ThisRow = StartRow To EndRow
      TotalWeight1 = TotalWeight1 + Sheets(InputSheet).Cells(ThisRow,
StartColumn + 9)
      TotalWeight2 = TotalWeight2 + Sheets(InputSheet).Cells(ThisRow,
StartColumn + 11)
  Next

   MeanWeight1 = TotalWeight1 / RequiredSample
   MeanWeight2 = TotalWeight2 / RequiredSample

  Sheets(InputSheet).Cells(StartRow - 7, StartColumn + 11) = "Case 1"
  Sheets(InputSheet).Cells(StartRow - 6, StartColumn + 11) = "Case 2"
  Sheets(InputSheet).Cells(StartRow - 7, StartColumn + 12) = " Mean Weight"
  Sheets(InputSheet).Cells(StartRow - 6, StartColumn + 12) = " Mean Weight"
  Sheets(InputSheet).Cells(StartRow - 7, StartColumn + 13) = MeanWeight1
  Sheets(InputSheet).Cells(StartRow - 6, StartColumn + 13) = MeanWeight2

 ' Creates a weight' column for case 1
   For ThisRow = StartRow To EndRow
```

```
        If Sheets(InputSheet).Cells(ThisRow, StartColumn + 9) > 0 Then
            Sheets(InputSheet).Cells(ThisRow, StartColumn + 13) =
Sheets(InputSheet).Cells(ThisRow, StartColumn + 9) / MeanWeight1
        End If

    Next

    ' Creates a weight' column for case 2
    For ThisRow = StartRow To EndRow
        If Sheets(InputSheet).Cells(ThisRow, StartColumn + 11) > 0 Then
            Sheets(InputSheet).Cells(ThisRow, StartColumn + 14) =
Sheets(InputSheet).Cells(ThisRow, StartColumn + 11) / MeanWeight2
        End If
    Next

    Sheets(InputSheet).Cells(StartRow - 8, StartingColumn + 1) = HighCount
    Sheets(InputSheet).Cells(StartRow - 7, StartingColumn + 1) = LowCount

    ' Outputs the weighted mean income for Case 1 and Case 2
    Sheets(InputSheet).Cells(StartRow - 10, StartColumn + 7) = "Income"
    Sheets(InputSheet).Cells(StartRow - 9, StartColumn + 7) = "Case 2 - Income"

    ' Weighted Mean Income = Sum(case weight*income)/Sum(case weight)
    Sheets(InputSheet).Cells(StartRow - 10, StartColumn + 8) =
Application.WorksheetFunction.SumProduct(Sheets(InputSheet).Range(Cells(Sta
rtRow, StartColumn + 6), Cells(StartRow + dpCount - 1, StartColumn + 6)),
Sheets(InputSheet).Range(Cells(StartRow, StartColumn + 9), Cells(StartRow +
dpCount - 1, StartColumn + 9))) /
Application.WorksheetFunction.Sum(Sheets(InputSheet).Range(Cells(StartRow,
StartColumn + 9), Cells(StartRow + dpCount - 1, StartColumn + 9)))
    Sheets(InputSheet).Cells(StartRow - 9, StartColumn + 8) =
Application.WorksheetFunction.SumProduct(Sheets(InputSheet).Range(Cells(Sta
rtRow, StartColumn + 6), Cells(StartRow + dpCount - 1, StartColumn + 6)),
Sheets(InputSheet).Range(Cells(StartRow, StartColumn + 11), Cells(StartRow +
dpCount - 1, StartColumn + 11))) /
Application.WorksheetFunction.Sum(Sheets(InputSheet).Range(Cells(StartRow,
StartColumn + 11), Cells(StartRow + dpCount - 1, StartColumn + 11)))

    ' Outputs the weighted mean gun ownership for Case 1 and Case 2
    Sheets(InputSheet).Cells(StartRow - 7, StartColumn + 7) = "Gun O/S"
    Sheets(InputSheet).Cells(StartRow - 6, StartColumn + 7) = "Case 2 - Gun O/S"

    ' Weighted Mean Gun o/s = Sum(case weight*gun)/Sum(case weight)
```

```
    Sheets(InputSheet).Cells(StartRow - 7, StartColumn + 8) =
Application.WorksheetFunction.SumProduct(Sheets(InputSheet).Range(Cells(Sta
rtRow, StartColumn + 7), Cells(StartRow + dpCount - 1, StartColumn + 7)),
Sheets(InputSheet).Range(Cells(StartRow, StartColumn + 9), Cells(StartRow +
dpCount - 1, StartColumn + 9))) /
Application.WorksheetFunction.Sum(Sheets(InputSheet).Range(Cells(StartRow,
StartColumn + 9), Cells(StartRow + dpCount - 1, StartColumn + 9)))
    Sheets(InputSheet).Cells(StartRow - 6, StartColumn + 8) =
Application.WorksheetFunction.SumProduct(Sheets(InputSheet).Range(Cells(Sta
rtRow, StartColumn + 7), Cells(StartRow + dpCount - 1, StartColumn + 7)),
Sheets(InputSheet).Range(Cells(StartRow, StartColumn + 11), Cells(StartRow +
dpCount - 1, StartColumn + 11))) /
Application.WorksheetFunction.Sum(Sheets(InputSheet).Range(Cells(StartRow,
StartColumn + 11), Cells(StartRow + dpCount - 1, StartColumn + 11)))

    ' At each row (`thisRow`), the weighted standard error for income and gun
ownership is calculated BY CASE, and summed with the results found in the
previous rows.
    ' The numerator and denominator calculation for the standard deviation portion
of the standard error calculation is as below.
    ' In addition, at each row the code also compares that row's DP coordinates
against the DP coordinates of the previous rows to determine whether any overlap
(overlap occurs if centres of both are less than 2*radius apart)

    LoopStartRow = StartRow
    LoopEndRow = StartRow + dpCount - 1 - 1

    stdDevNum(1, 1) = 0
    stdDevNum(1, 2) = 0
    stdDevNum(2, 1) = 0
    stdDevNum(2, 2) = 0

    stdDevDen(1) = 0
    stdDevDen(2) = 0

    For ThisRow = LoopStartRow To LoopEndRow
        For ThisCase = 1 To 2

            stdDevNum(ThisCase, 1) = stdDevNum(ThisCase, 1) +
Sheets(InputSheet).Cells(ThisRow, StartColumn + 13 + (ThisCase - 1) * 1) *
((Sheets(InputSheet).Cells(ThisRow, StartColumn + 6) -
Sheets(InputSheet).Cells(StartRow - 10 + (ThisCase - 1) * 1, StartColumn + 8)) ^
2)
```

```
        stdDevNum(ThisCase, 2) = stdDevNum(ThisCase, 2) +
Sheets(InputSheet).Cells(ThisRow, StartColumn + 13 + (ThisCase - 1) * 1) *
((Sheets(InputSheet).Cells(ThisRow, StartColumn + 7) -
Sheets(InputSheet).Cells(StartRow - 7 + (ThisCase - 1) * 1, StartColumn + 8)) ^
2)

        stdDevDen(ThisCase) = stdDevDen(ThisCase) +
(Sheets(InputSheet).Cells(ThisRow, StartColumn + 13 + (ThisCase - 1) * 1))
      Next

      ' Outputs the standard deviation for Case 1 on the input sheet

     Sheets(InputSheet).Cells(StartRow - 10, StartColumn + 10) =
Sqr(stdDevNum(1, 1) / (stdDevDen(1) - 1))
     Sheets(InputSheet).Cells(StartRow - 7, StartColumn + 10) =
Sqr(stdDevNum(1, 2) / (stdDevDen(1) - 1))

     ' Prints out the datapoints which overlap with the DP being sampled (in
`thisRow`)
     For CurrentRow = StartRow To ThisRow - 1

        hDistanceDP = Abs(Sheets(InputSheet).Cells(CurrentRow, StartColumn +
2) - Sheets(InputSheet).Cells(ThisRow, StartColumn + 2))
        vDistanceDP = Abs(Sheets(InputSheet).Cells(CurrentRow, StartColumn +
3) - Sheets(InputSheet).Cells(ThisRow, StartColumn + 3))

        If Sqr(hDistanceDP ^ 2 + vDistanceDP ^ 2) < 2 * BuildingRadius Then
          ' If there`s already info of an overlapping DP in the cell, the cell won`t
be empty and the info of the additional overlapping DP will just added to the cell
          If Sheets(InputSheet).Cells(ThisRow, StartColumn + 12) <> "" Then
            Sheets(InputSheet).Cells(ThisRow, StartColumn + 12) =
Sheets(InputSheet).Cells(ThisRow, StartColumn + 12) & ", " &
Sheets(InputSheet).Cells(CurrentRow, StartColumn)
          End If

          ' If this is the first overlapping DP, the cell will be empty and the DP`s
info will be entered in it
          If Sheets(InputSheet).Cells(ThisRow, StartColumn + 12) = "" Then
            Sheets(InputSheet).Cells(ThisRow, StartColumn + 12) =
Sheets(InputSheet).Cells(CurrentRow, StartColumn)
          End If
        End If
      Next
    Next
```

```
    ' Outputs the standard error for the two cases on the input sheet
    Sheets(InputSheet).Cells(StartRow - 10, StartColumn + 9) =
(Sqr(stdDevNum(1, 1) / (stdDevDen(1) - 1)) / Sqr(RequiredSample)) * Sqr(1 -
RequiredSample / PlacedBuildings)
    Sheets(InputSheet).Cells(StartRow - 9, StartColumn + 9) = (Sqr(stdDevNum(2,
1) / (stdDevDen(2) - 1)) / Sqr(RequiredSample))

    Sheets(InputSheet).Cells(StartRow - 7, StartColumn + 9) = (Sqr(stdDevNum(1,
2) / (stdDevDen(1) - 1)) / Sqr(RequiredSample - 1)) * Sqr(1 - RequiredSample /
PlacedBuildings)
    Sheets(InputSheet).Cells(StartRow - 6, StartColumn + 9) = (Sqr(stdDevNum(2,
2) / (stdDevDen(2) - 1)) / Sqr(RequiredSample))

    ' Outputs the statistics on the outputSheet
    Sheets(OutputSheet).Select
    Sheets(OutputSheet).Cells(ThisIteration + 4, 1) = ThisIteration

    ' ITERATION STATS ON STATISTICAL OUTPUT SHEET

    ' Case 1: Income
    Sheets(OutputSheet).Cells(ThisIteration + 4, 2) =
Sheets(InputSheet).Cells(StartRow - 10, StartColumn + 8)
    Sheets(OutputSheet).Cells(ThisIteration + 4, 3) =
Sheets(InputSheet).Cells(StartRow - 10, StartColumn + 9)
    Sheets(OutputSheet).Cells(ThisIteration + 4, 4) =
Sheets(InputSheet).Cells(StartRow - 10, StartColumn + 9) *
Sheets(InputSheet).Cells(StartRow - 10, StartColumn + 9)

    ' Case 1: Gun Ownership
    Sheets(OutputSheet).Cells(ThisIteration + 4, 5) =
Sheets(InputSheet).Cells(StartRow - 7, StartColumn + 8)
    Sheets(OutputSheet).Cells(ThisIteration + 4, 6) =
Sheets(InputSheet).Cells(StartRow - 7, StartColumn + 9)
    Sheets(OutputSheet).Cells(ThisIteration + 4, 7) =
Sheets(InputSheet).Cells(StartRow - 7, StartColumn + 9) *
Sheets(InputSheet).Cells(StartRow - 7, StartColumn + 9)

    ' Case 2: Income
    Sheets(OutputSheet).Cells(ThisIteration + 4, 8) =
Sheets(InputSheet).Cells(StartRow - 9, StartColumn + 8)
    Sheets(OutputSheet).Cells(ThisIteration + 4, 9) =
Sheets(InputSheet).Cells(StartRow - 9, StartColumn + 9)
```

```
    ' Case 2: Gun Ownership
    Sheets(OutputSheet).Cells(ThisIteration + 4, 10) =
Sheets(InputSheet).Cells(StartRow - 6, StartColumn + 8)
    Sheets(OutputSheet).Cells(ThisIteration + 4, 11) =
Sheets(InputSheet).Cells(StartRow - 6, StartColumn + 9)

    ' Outputs the statistics on the outputSheet
    Sheets(IterationSheet).Select
    Sheets(IterationSheet).Cells(IterationOutputRow, PasteToCol) = ThisIteration

    ' ITERATION STATS ON ITERATION OUTPUT SHEET

    ' Copy sample information for each iteration from Coordinates Sheet and paste
to Output Sheet
    Sheets(InputSheet).Select

    Range(Cells(StartRow, StartColumn), Cells(StartRow, StartColumn +
14)).Select
    Selection.Copy

    Range(Cells(StartRow + 1, StartColumn), Cells(EndRow, StartColumn +
14)).Select
    Selection.PasteSpecial Paste:=xlPasteFormats, Operation:=xlNone, _
    SkipBlanks:=False, Transpose:=False

    'Building Selected Details Calculation

    Sheets(InputSheet).Select

    Cells(StartRow + 1, StartingColumn).Select

    If Cells(StartRow + 1, StartingColumn) <> "" Then

        Range(Selection, Selection.End(xlToRight)).Select
        Range(Selection, Selection.End(xlDown)).Select
        Selection.Clear

    End If

    Range(Cells(StartRow, StartingColumn), Cells(StartRow, StartingColumn +
2)).Select
    Selection.Copy
    Range(Cells(StartRow + 1, StartingColumn), Cells(EndRow, StartingColumn +
2)).Select
```

```
   ActiveSheet.Paste

   Set BuildingNumberRange = Range(Cells(StartRow, StartingColumn),
Cells(EndRow, StartingColumn))
   Set BuildingXCoordinate = Range(Cells(StartRow, StartingColumn + 1),
Cells(EndRow, StartingColumn + 1))
   Set BuildingIncome = Range(Cells(StartRow, StartingColumn + 2),
Cells(EndRow, StartingColumn + 2))

   ActiveWorkbook.Names.Add Name:="BuildingNumberRange",
RefersTo:=BuildingNumberRange
   ActiveWorkbook.Names.Add Name:="BuildingXCoordinate",
RefersTo:=BuildingXCoordinate
   ActiveWorkbook.Names.Add Name:="BuildingIncome",
RefersTo:=BuildingIncome

   Sheets(InputSheet).Range(Cells(StartRow, StartColumn), Cells(StartRow +
dpCount, StartColumn + 18)).Copy
   Sheets(IterationSheet).Select
   Sheets(IterationSheet).Cells(IterationOutputRow, PasteToCol + 1).Select
   Selection.PasteSpecial Paste:=xlPasteFormats, Operation:=xlNone, _
      SkipBlanks:=False, Transpose:=False
   Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
      :=False, Transpose:=False
   Application.CutCopyMode = False

   IterationOutputRow = IterationOutputRow + dpCount + 1

   If IterationOutputRow >= 1000000 Then
      IterationOutputRow = 4
      PasteToCol = PasteToCol + 21
   End If


   sngEnd = Timer                          ' Get end time.
   sngElapsed = Format(sngElapsed + sngEnd - sngStart, "Fixed") ' Elapsed time.


   PctDone = ThisIteration / IterationCount

   PctDone = ThisIteration / IterationCount

   If ShowFormFlag = 1 Then
```

```
        With UserForm1
        .LabelProgress.BackColor = RGB(0, 70, 127)
        .Label1.Caption = "Iteration Number " & ThisIteration & " of " &
IterationCount
        .Label2.Caption = "Time Elapsed: " & TimeString(sngElapsed, True)
        TimeRemaining = Format((IterationCount - ThisIteration) * sngElapsed /
ThisIteration, "Fixed")
        .Label3.Caption = "Estimated Time Remaining: " &
TimeString(TimeRemaining, True)
        .FrameProgress.Caption = Format(PctDone, "0%")
        .LabelProgress.Width = PctDone * (.FrameProgress.Width - 10)
        .Repaint
        End With
        DoEvents

    End If

Next

If ShowFormFlag = 1 Then

    Unload UserForm1

End If

' SUMMARY STATS

' Define required rows and columns
Sheets(OutputSheet).Select

OutputSheetOutputRow = 5
OutputSheetEndRow = OutputSheetOutputRow + IterationCount - 1
OutputSheetOutputCol = 13

Case1IncomeAverageCol = 2
Case1IncomeSErrorCol = 3
Case1IncomeSErrorSqCol = 4
Case1GunOwnershipAverageCol = 5
Case1GunOwnershipSErrorCol = 6
Case1GunOwnershipSErrorSqCol = 7

Case2IncomeAverageCol = 8
Case2IncomeSErrorCol = 9
Case2GunOwnershipAverageCol = 10
```

```
Case2GunOwnershipSErrorCol = 11

' Define required ranges
Set Case1IncomeAverageRange = Range(Cells(OutputSheetOutputRow,
Case1IncomeAverageCol), Cells(OutputSheetEndRow,
Case1IncomeAverageCol))
Set Case1IncomeSErrorRange = Range(Cells(OutputSheetOutputRow,
Case1IncomeSErrorCol), Cells(OutputSheetEndRow, Case1IncomeSErrorCol))
Set Case1IncomeSErrorSqRange = Range(Cells(OutputSheetOutputRow,
Case1IncomeSErrorSqCol), Cells(OutputSheetEndRow,
Case1IncomeSErrorSqCol))
Set Case1GunOwnershipAverageRange = Range(Cells(OutputSheetOutputRow,
Case1GunOwnershipAverageCol), Cells(OutputSheetEndRow,
Case1GunOwnershipAverageCol))
Set Case1GunOwnershipSErrorRange = Range(Cells(OutputSheetOutputRow,
Case1GunOwnershipSErrorCol), Cells(OutputSheetEndRow,
Case1GunOwnershipSErrorCol))
Set Case1GunOwnershipSErrorSqRange = Range(Cells(OutputSheetOutputRow,
Case1GunOwnershipSErrorSqCol), Cells(OutputSheetEndRow,
Case1GunOwnershipSErrorSqCol))

Set Case2IncomeAverageRange = Range(Cells(OutputSheetOutputRow,
Case2IncomeAverageCol), Cells(OutputSheetEndRow,
Case2IncomeAverageCol))
Set Case2IncomeSErrorRange = Range(Cells(OutputSheetOutputRow,
Case2IncomeSErrorCol), Cells(OutputSheetEndRow, Case2IncomeSErrorCol))
Set Case2GunOwnershipAverageRange = Range(Cells(OutputSheetOutputRow,
Case2GunOwnershipAverageCol), Cells(OutputSheetEndRow,
Case2GunOwnershipAverageCol))
Set Case2GunOwnershipSErrorRange = Range(Cells(OutputSheetOutputRow,
Case2GunOwnershipSErrorCol), Cells(OutputSheetEndRow,
Case2GunOwnershipSErrorCol))

ActiveWorkbook.Names.Add Name:="Case1IncomeAverageRange",
RefersTo:=Case1IncomeAverageRange
ActiveWorkbook.Names.Add Name:="Case1IncomeSErrorRange",
RefersTo:=Case1IncomeSErrorRange
ActiveWorkbook.Names.Add Name:="Case1IncomeSErrorSqRange",
RefersTo:=Case1IncomeSErrorSqRange
ActiveWorkbook.Names.Add Name:="Case1GunOwnershipAverageRange",
RefersTo:=Case1GunOwnershipAverageRange
ActiveWorkbook.Names.Add Name:="Case1GunOwnershipSErrorRange",
RefersTo:=Case1GunOwnershipSErrorRange
```

```
ActiveWorkbook.Names.Add Name:="Case1GunOwnershipSErrorSqRange",
RefersTo:=Case1GunOwnershipSErrorSqRange

ActiveWorkbook.Names.Add Name:="Case2IncomeAverageRange",
RefersTo:=Case2IncomeAverageRange
ActiveWorkbook.Names.Add Name:="Case2IncomeSErrorRange",
RefersTo:=Case2IncomeSErrorRange
ActiveWorkbook.Names.Add Name:="Case2GunOwnershipAverageRange",
RefersTo:=Case2GunOwnershipAverageRange
ActiveWorkbook.Names.Add Name:="Case2GunOwnershipSErrorRange",
RefersTo:=Case2GunOwnershipSErrorRange

Cells(OutputSheetOutputRow, OutputSheetOutputCol).FormulaR1C1 =
"=AVERAGE(Case1IncomeAverageRange)"
Cells(OutputSheetOutputRow, OutputSheetOutputCol + 1).FormulaR1C1 =
"=Stdev(Case1IncomeAverageRange)"
Cells(OutputSheetOutputRow, OutputSheetOutputCol + 2).FormulaR1C1 =
"=sqrt(AVERAGE(Case1IncomeSErrorSqRange))"

Cells(OutputSheetOutputRow, OutputSheetOutputCol + 3).FormulaR1C1 =
"=AVERAGE(Case1GunOwnershipAverageRange)"
Cells(OutputSheetOutputRow, OutputSheetOutputCol + 4).FormulaR1C1 =
"=Stdev(Case1GunOwnershipAverageRange)"
Cells(OutputSheetOutputRow, OutputSheetOutputCol + 5).FormulaR1C1 =
"=sqrt(AVERAGE(Case1GunOwnershipSErrorSqRange))"

Cells(OutputSheetOutputRow, OutputSheetOutputCol + 6).FormulaR1C1 =
"=AVERAGE(Case2IncomeAverageRange)"
Cells(OutputSheetOutputRow, OutputSheetOutputCol + 7).FormulaR1C1 =
"=Stdev(Case2IncomeAverageRange)"
Cells(OutputSheetOutputRow, OutputSheetOutputCol + 8).FormulaR1C1 =
"=AVERAGE(Case2IncomeSErrorRange)"

Cells(OutputSheetOutputRow, OutputSheetOutputCol + 9).FormulaR1C1 =
"=AVERAGE(Case2GunOwnershipAverageRange)"
Cells(OutputSheetOutputRow, OutputSheetOutputCol + 10).FormulaR1C1 =
"=Stdev(Case2GunOwnershipAverageRange)"
Cells(OutputSheetOutputRow, OutputSheetOutputCol + 11).FormulaR1C1 =
"=AVERAGE(Case2GunOwnershipSErrorRange)"

'Puts a comma in all columns 2-9 on Statistical Output sheet
Sheets(OutputSheet).Range(Cells(5, 2), Cells(ThisIteration + 4 - 1, 9)).Style =
"Comma"
```

```
'Bolds and centres font in column 1 on Statistical Output sheet
Sheets(OutputSheet).Range(Cells(5, 1), Cells(ThisIteration + 4 - 1, 1)).Select
  Selection.Font.Bold = True
    With Selection
      .HorizontalAlignment = xlCenter
   End With

' Last Instructions for the Sub
Sheets("Iteration Output").Select
Columns("A:BZ").Select
Columns("A:BZ").EntireColumn.AutoFit
Range("A4").Select
Columns("J:K").Select
Selection.ColumnWidth = 10.71

' Hide Case 2 Columns of Probs and Weights

Sheets("Iteration Output").Select
Columns("L:M").Select
Selection.EntireColumn.Hidden = True
Columns("P:P").Select
Selection.EntireColumn.Hidden = True
Range("A1").Select

Sheets("Statistical Output").Select
Columns("A:BZ").Select
Columns("A:BZ").EntireColumn.AutoFit

Columns("H:K").Select
Selection.EntireColumn.Hidden = True

Columns("S:X").Select
Selection.EntireColumn.Hidden = True

Range("A5").Select

On Error GoTo 0
Application.DisplayAlerts = True
Application.ScreenUpdating = True

Sheets(InputSheet).Activate
Cells(1, 1).Select

Application.CutCopyMode = False
```

```
ActiveWorkbook.Save

End Sub
```

iii. Time Functions Module

This module was used to convert the estimated seconds required to run all iterations (which was calculated in Module Sample) into hours, minutes and seconds.

```
Public Function TimeString(Seconds As Long, Optional Verbose _
As Boolean = False) As String

'if verbose = false, returns
'something like
'02:22.08
'if true, returns
'2 hours, 22 minutes, and 8 seconds

Dim lHrs As Long
Dim lMinutes As Long
Dim lSeconds As Long

lSeconds = Seconds

lHrs = Int(lSeconds / 3600)
lMinutes = (Int(lSeconds / 60)) - (lHrs * 60)
lSeconds = Int(lSeconds Mod 60)

Dim sAns As String


If lSeconds = 60 Then
   lMinutes = lMinutes + 1
   lSeconds = 0
End If

If lMinutes = 60 Then
   lMinutes = 0
   lHrs = lHrs + 1
End If
```

```
sAns = Format(CStr(lHrs), "#####0") & ":" & _
  Format(CStr(lMinutes), "00") & "." & _
  Format(CStr(lSeconds), "00")

If Verbose Then sAns = TimeStringtoEnglish(sAns)
TimeString = sAns

End Function

Private Function TimeStringtoEnglish(sTimeString As String) _
As String

Dim sAns As String
Dim sHour, sMin As String, sSec As String
Dim iTemp As Integer, sTemp As String
Dim iPos As Integer
iPos = InStr(sTimeString, ":") - 1

sHour = Left$(sTimeString, iPos)
If CLng(sHour) <> 0 Then
   sAns = CLng(sHour) & " hour"
   If CLng(sHour) > 1 Then sAns = sAns & "s"
   sAns = sAns & ", "
End If

sMin = Mid$(sTimeString, iPos + 2, 2)

iTemp = sMin

If sMin = "00" Then
  sAns = IIf(Len(sAns), sAns & "0 minutes, and ", "")
Else
  sTemp = IIf(iTemp = 1, " minute", " minutes")
  sTemp = IIf(Len(sAns), sTemp & ", and ", sTemp & " and ")
  sAns = sAns & Format$(iTemp, "##") & sTemp
End If

iTemp = Val(Right$(sTimeString, 2))
sSec = Format$(iTemp, "#0")
sAns = sAns & sSec & " second"
If iTemp <> 1 Then sAns = sAns & "s"

TimeStringtoEnglish = sAns
```

End Function

iv. Show Dialogue Module

This module was used to create a prompt box which would first ask the user how many iterations they wanted (i.e. number of samples of size n from the population) and then ran the code for that number of iterations.

```
Sub ShowDialog()

'Asks for number of iterations required by user (each iteration = a sample of
'Sample Count' observations)
IterationCountStr = InputBox("Please specify how many iterations are required.",
"Number of Iterations", "100")

ShowFormFlag = 1

If ShowFormFlag = 1 Then

   UserForm1.LabelProgress.Width = 0
   UserForm1.Show

Else
   Call Main
End If

End Sub

Public Sub Main()

   Call sample

   If ShowFormFlag = 1 Then

      Unload UserForm1

   End If
End Sub

Public Sub UpdateProgress(Pct)
   LabelProgress.BackColor = RGB(0, 70, 127)
   Label1.Caption = "Iteration Number " & ThisIteration & " of " &
IterationCount
```

```
   Label2.Caption = "Time Elapsed: " & TimeString(sngElapsed, True)
   TimeRemaining = Format((IterationCount - ThisIteration) * sngElapsed /
ThisIteration, "Fixed")
   Label3.Caption = "Estimated Time Remaining: " &
TimeString(TimeRemaining, True)
   PctDone = ThisIteration / IterationCount

   With UserForm1
   .FrameProgress.Caption = Format(Pct, "0%")
   .LabelProgress.Width = Pct * (.FrameProgress.Width - 10)
   .Repaint
   End With
   DoEvents
End Sub
```

v. Time Functions Module

This module was used to combine information for multiple sets of iterations, for comparison amongst the estimates.

```
Public Sub MasterOutputMaker()

Application.ScreenUpdating = False
Application.DisplayAlerts = False

Dim HomeSheet As String
HomeSheet = "Master Output"
Sheets(HomeSheet).Activate

Dim FolderLocation As String
FolderLocation = Range("FolderLoc").Value

Dim StartFileLocRow As Integer
Dim StartFileLocCol As Integer

StartFileLocRow = Range("StartFileLoc").Row
StartFileLocCol = Range("StartFileLoc").Column

Dim CheckFile As String

CurrentFileRow = StartFileLocRow
CurrentFileCol = StartFileLocCol
```

```
CheckFile = Cells(CurrentFileRow, CurrentFileCol + 1) & ".xlsm"
CurrentWorkbook = ActiveWorkbook.Name

Dim OutputFileLocRow As Integer
Dim OutputFileLocCol As Integer

OutputFileLocRow = Range("OutputCell").Row
OutputFileLocCol = Range("OutputCell").Column

CurrentOutputRow = OutputFileLocRow

Range("OutputCell").Select

Dim SecondRangeStartCol As Integer
Dim ThirdRangeStartCol As Integer
Dim FourthRangeStartCol As Integer
Dim FifthRangeStartCol As Integer

SecondRangeStartCol = Range("SecondRangeStart").Column
ThirdRangeStartCol = Range("ThirdRangeStart").Column
FourthRangeStartCol = Range("FourthRangeStart").Column
FifthRangeStartCol = Range("FifthRangeStart").Column

If Range("OutputCell") = "" Then

Else

   Range("OutputCell").Select
   Range(Selection, ActiveCell.SpecialCells(xlLastCell)).Select
   Selection.Clear

End If

Do

   If CheckFile = ".xlsm" Then

      Exit Do

   Else
      FileToOpen = CheckFile
      Cells(CurrentOutputRow, OutputFileLocCol) = CheckFile

   End If
```

```
    Workbooks.Open Filename:=FolderLocation & CheckFile

    NumItr = Range("ItrCount").Value

    ' Copy Output from Source File
    Sheets("Statistical Output").Select
    Range("SumStatsOutput").Select
    Selection.Copy
    Windows(CurrentWorkbook).Activate

    Cells(CurrentOutputRow, OutputFileLocCol + 6).Select
    Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
        :=False, Transpose:=False

    Cells(CurrentOutputRow, OutputFileLocCol + 1).Value = NumItr
    Cells(CurrentOutputRow, OutputFileLocCol + 2).Value =
Range("IncomeMean").Value
    Cells(CurrentOutputRow, OutputFileLocCol + 3).Value =
Range("GunProportion").Value
    FPC_Value = Sqr(1 - (Range("SampleSize") / (Range("NumLowBuildings") +
Range("NumHighBuildings"))))
    Cells(CurrentOutputRow, OutputFileLocCol + 4).Value =
Range("IncomeStdDev") / Sqr(Range("SampleSize")) * FPC_Value
    Cells(CurrentOutputRow, OutputFileLocCol + 5).Value =
Range("GunStdDev") / Sqr(Range("SampleSize")) * FPC_Value

    'Selection.PasteSpecial Paste:=xlPasteFormats, Operation:=xlNone, _
    '    SkipBlanks:=False, Transpose:=False

    ' Copy Second Range
    Windows(CheckFile).Activate
    Sheets("Analysis of Results").Select
    Range("DiffSDMeanSE").Select
    Selection.Copy
    Windows(CurrentWorkbook).Activate

    Cells(CurrentOutputRow, SecondRangeStartCol).Select
    ActiveSheet.Paste
    Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
        :=False, Transpose:=False

    ' Copy Third Range
    Windows(CheckFile).Activate
```

```
    Sheets("Analysis of Results").Select
    Range("PropSDMeanSE").Select
    Selection.Copy
    Windows(CurrentWorkbook).Activate

    Cells(CurrentOutputRow, ThirdRangeStartCol).Select
    ActiveSheet.Paste
    Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
       :=False, Transpose:=False

    ' Copy Fourth Range
    Windows(CheckFile).Activate
    Sheets("Analysis of Results").Select
    Range("DiffSEMeanSE").Select
    Selection.Copy
    Windows(CurrentWorkbook).Activate

    Cells(CurrentOutputRow, FourthRangeStartCol).Select
    ActiveSheet.Paste
    Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
       :=False, Transpose:=False

    ' Copy Fifth Range
    Windows(CheckFile).Activate
    Sheets("Analysis of Results").Select
    Range("PropSEMeanSE").Select
    Selection.Copy
    Windows(CurrentWorkbook).Activate

    Cells(CurrentOutputRow, FifthRangeStartCol).Select
    ActiveSheet.Paste
    Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
       :=False, Transpose:=False

    Application.CutCopyMode = False
    Windows(CheckFile).Close
    Range("A1").Select

    CurrentFileRow = CurrentFileRow + 1
    CurrentOutputRow = CurrentOutputRow + 1
    CheckFile = Cells(CurrentFileRow, CurrentFileCol + 1) & ".xlsm"

Loop
```

```
'Formatting
Range("OutputCell").Select
Range(Selection, ActiveCell.SpecialCells(xlLastCell)).Select
Selection.Borders(xlDiagonalDown).LineStyle = xlNone
Selection.Borders(xlDiagonalUp).LineStyle = xlNone
Selection.Borders(xlEdgeLeft).LineStyle = xlNone
Selection.Borders(xlEdgeBottom).LineStyle = xlNone
Selection.Borders(xlEdgeRight).LineStyle = xlNone
Selection.Borders(xlInsideVertical).LineStyle = xlNone
Selection.Borders(xlInsideHorizontal).LineStyle = xlNone

Range("OutputCell").Select

Application.ScreenUpdating = True
Application.DisplayAlerts = True

End Sub
```